



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/453,682	01/25/2022	11233780	63121/0184	2186

1912 7590 01/05/2022

AMSTER, ROTHSTEIN & EBENSTEIN LLP
90 PARK AVENUE
NEW YORK, NY 10016

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (application filed on or after May 29, 2000)

The Patent Term Adjustment is 29 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

INVENTOR(s) (Please see PAIR WEB site <http://pair.uspto.gov> for additional inventors):

John A. Nix, Evanston, IL;

APPLICANT(s) (Please see PAIR WEB site <http://pair.uspto.gov> for additional applicants):

Network-1 Technologies, Inc., New York, NY;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

Doc code: IDS

PTO/SB/08a (02-18)

Doc description: Information Disclosure Statement (IDS) Filed

Approved for use through 11/30/2020. OMB 0651-0031

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		16453682
	Filing Date		2019-06-26
	First Named Inventor		John A. Nix
	Art Unit		2432
	Examiner Name	HO, Dao Q.	
	Attorney Docket Number		63121/0184

U.S.PATENTS						Remove
Examiner Initial*	Cite No	Patent Number	Kind Code ¹	Issue Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear
	1	9807605	B2	2017-10-31	Gao et al.	
	2	8577337	B2	2013-11-05	O'Leary	
Change(s) applied to document, /M.H.E./ 7/2/2021	3	9800993	B2	2013-11-05 10/2017	Lee et al.	
	4	9794775	B2	2017-10-17	Huber et al.	
	5	9674690	B2	2017-06-06	Lee et al.	
	6	9628981	B2	2017-04-18	Park et al.	
	7	9572016	B2	2017-02-14	Cormier et al.	
	8	9332129	B2	2016-05-03	Li et al.	

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

1912 7590 09/14/2021
AMSTER, ROTHSTEIN & EBENSTEIN LLP
90 PARK AVENUE
NEW YORK, NY 10016

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

Albert J. Boardman	(Typed or printed name)
/Albert J. Boardman/	(Signature)
December 10, 2021	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/453,682	06/26/2019	John A. Nix	63121/0184	2186

TITLE OF INVENTION: Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$600	\$0.00	\$0.00	\$600	12/14/2021

EXAMINER	ART UNIT	CLASS-SUBCLASS
HO, DAO Q	2432	713-171000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-09 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

- (1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
- (2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____
2 Amster, Rothstein & Ebenstein LLP
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Network-1 Technologies, Inc.

New York, NY

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. Fees submitted: ☒ Issue Fee ☐ Publication Fee (if required) ☐ Advance Order - # of Copies _____

4b. Method of Payment: (Please first reapply any previously paid fee shown above)

☒ Electronic Payment via EFS-Web ☐ Enclosed check ☐ Non-electronic payment by credit card (Attach form PTO-2038)

☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. 011785

5. Change in Entity Status (from status indicated above)

- ☐ Applicant certifying micro entity status. See 37 CFR 1.29
- ☐ Applicant asserting small entity status. See 37 CFR 1.27
- ☐ Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature /Albert J. Boardman/

Date December 10, 2021

Typed or printed name Albert J. Boardman

Registration No. 70,601

Electronic Patent Application Fee Transmittal

Application Number:	16453682			
Filing Date:	26-Jun-2019			
Title of Invention:	Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication			
First Named Inventor/Applicant Name:	John A. Nix			
Filer:	Albert J. Boardman/Victoria Gilmore			
Attorney Docket Number:	63121/0184			
Filed as Small Entity				
Filing Fees for Utility under 35 USC 111(a)				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
PUBL. FEE- EARLY, VOLUNTARY, OR NORMAL	1504	1	0	0
UTILITY APPL ISSUE FEE	2501	1	600	600
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				600

Electronic Acknowledgement Receipt

EFS ID:	44489696
Application Number:	16453682
International Application Number:	
Confirmation Number:	2186
Title of Invention:	Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication
First Named Inventor/Applicant Name:	John A. Nix
Customer Number:	1912
Filer:	Albert J. Boardman/Victoria Gilmore
Filer Authorized By:	Albert J. Boardman
Attorney Docket Number:	63121/0184
Receipt Date:	10-DEC-2021
Filing Date:	26-JUN-2019
Time Stamp:	16:34:45
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	\$ 600
RAM confirmation Number	E2021B0G35012382
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

File Listing:					
Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	184-PTOL-85.pdf	97279	no	1
			73f40d3ebd5bab505ebc558aa0a7c40ebd826042		
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	40650	no	2
			dc0202f3e2682a386b998dba12f52f4aab3116471		
Warnings:					
Information:					
Total Files Size (in bytes):			137929		
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

1912 7590 09/14/2021
AMSTER, ROTHSTEIN & EBENSTEIN LLP
90 PARK AVENUE
NEW YORK, NY 10016

EXAMINER

HO, DAO Q

ART UNIT

PAPER NUMBER

2432

DATE MAILED: 09/14/2021

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/453,682	06/26/2019	John A. Nix	63121/0184	2186

TITLE OF INVENTION: Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$600	\$0.00	\$0.00	\$600	12/14/2021

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

1912 7590 09/14/2021
AMSTER, ROTHSTEIN & EBENSTEIN LLP
90 PARK AVENUE
NEW YORK, NY 10016

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

(Typed or printed name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/453,682	06/26/2019	John A. Nix	63121/0184	2186

TITLE OF INVENTION: Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$600	\$0.00	\$0.00	\$600	12/14/2021

EXAMINER	ART UNIT	CLASS-SUBCLASS
HO, DAO Q	2432	713-171000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-09 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

- (1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
- (2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. Fees submitted: ☐ Issue Fee ☐ Publication Fee (if required) ☐ Advance Order - # of Copies _____

4b. Method of Payment: (Please first reapply any previously paid fee shown above)

☐ Electronic Payment via EFS-Web ☐ Enclosed check ☐ Non-electronic payment by credit card (Attach form PTO-2038)

☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

5. Change in Entity Status (from status indicated above)

- ☐ Applicant certifying micro entity status. See 37 CFR 1.29
- ☐ Applicant asserting small entity status. See 37 CFR 1.27
- ☐ Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____

Date _____

Typed or printed name _____

Registration No. _____



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/453,682	06/26/2019	John A. Nix	63121/0184	2186
1912	7590	09/14/2021	EXAMINER	
AMSTER, ROTHSTEIN & EBENSTEIN LLP			HO, DAO Q	
90 PARK AVENUE			ART UNIT	PAPER NUMBER
NEW YORK, NY 10016			2432	
DATE MAILED: 09/14/2021				

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No. 16/453,682	Applicant(s) Nix, John A.	
	Examiner DAO Q HO	Art Unit 2432	AIA (FITF) Status Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 08/25/2021.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ____.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.

3. ☒ The allowed claim(s) is/are 1-13, 15-16 and 18-22. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov**.

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:
a) ☐ All b) ☐ Some* c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date ____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892) 2. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____. 3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material _____. 4. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____.	5. ☐ Examiner's Amendment/Comment 6. ☒ Examiner's Statement of Reasons for Allowance 7. ☐ Other _____.
--	--

/DAO Q HO/
Primary Examiner, Art Unit 2432

Notice of Pre-AIA or AIA Status

The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

DETAILED ACTION

Response to Amendment

This is a reply to the request for Continued Examination (RCE) filed on 08/25/2021, in which Claim(s) 1-13, 15-16 and 18-22 are presented for examination.

Claim(s) 14, 17 are cancelled.

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 08/25/2021 has been entered.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on 08/25/2021(2) is (are) being consider by the examiner.

Allowable Subject Matter

Claims 1-13, 15-16 and 18-22 are allowed.

The following is an examiner's statement of reasons for allowance:

This communication warrants No Examiner's Reason for Allowance as it was previously stated in the Notice of Allowance filed on 06/29/2021, as such the reasons for allowance are in all probability evident from the record and no statement is deemed necessary (see MPEP 1302.14).

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DAO Q HO whose telephone number is (571)270-5998. The examiner can normally be reached on 7:00am - 5:00pm.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Nickerson can be reached on (469) 295-9235. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

***/DAO Q HO/
Primary Examiner, Art Unit 2432***

Issue Classification 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

CPC						
Symbol					Type	Version
H04L	/	63	/	08	F	2013-01-01
H04L	/	9	/	0869	I	2013-01-01
H04L	/	63	/	0428	I	2013-01-01
H04L	/	9	/	0819	I	2013-01-01
H04L	/	9	/	3271	I	2013-01-01
H04W	/	4	/	70	I	2018-02-01
H04W	/	12	/	35	I	2021-01-01
H04L	/	63	/	062	I	2013-01-01
H04W	/	12	/	06	I	2013-01-01
H04B	/	1	/	3816	I	2013-01-01
H04L	/	63	/	0435	I	2013-01-01
H04L	/	63	/	06	I	2013-01-01
H04L	/	63	/	101	I	2013-01-01

CPC Combination Sets					
Symbol				Type	Set
	/		/		

NONE (Assistant Examiner)		Total Claims Allowed: 20	
/DAO Q HO/ Primary Examiner, Art Unit 2432 (Primary Examiner)		10 September 2021 (Date)	O.G. Print Claim(s) 1
			O.G. Print Figure 1

<i>Issue Classification</i> 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

INTERNATIONAL CLASSIFICATION			
CLAIMED			
H04L	/	29	/ 06
NON-CLAIMED			
	/		/

US ORIGINAL CLASSIFICATION	
CLASS	SUBCLASS

CROSS REFERENCES(S)						
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					

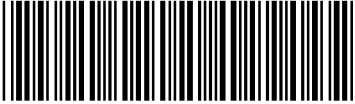
NONE		Total Claims Allowed:	
(Assistant Examiner)	(Date)	20	
/DAO Q HO/ Primary Examiner, Art Unit 2432	10 September 2021	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	1

Issue Classification 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47

CLAIMS															
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1		10		19										
	2		11		20										
	3		12		21										
	4		13		22										
	5		14												
	6		15												
	7		16												
	8		17												
	9		18												

NONE (Assistant Examiner) _____ (Date) _____		Total Claims Allowed: 20	
/DAO Q HO/ Primary Examiner, Art Unit 2432 (Primary Examiner) _____ (Date) 10 September 2021		O.G. Print Claim(s) 1	O.G. Print Figure 1

<i>Search Notes</i> 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

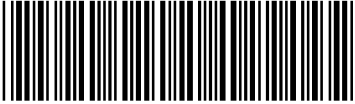
CPC - Searched*		
Symbol	Date	Examiner
H04L9/0819, 0869, 3247	01/14/2021	dh
H04L63/08, 428, 0853	01/14/2021	dh
H04W12/06	01/14/2021	dh
update	06/23/2021	dh
update	08/13/2021	dh
update	09/10/2021	dh

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

/DAO Q HO/ Primary Examiner, Art Unit 2432	
---	--

<i>Search Notes</i> 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

Search Notes		
Search Notes	Date	Examiner
see attached EAST search history	01/14/2021	dh
inventors and assignee search in EAST	01/14/2021	dh
NPL: subscription authenticating using SIM card with key exchange	01/14/2021	dh
CPC searches with delimiter	01/14/2021	dh
above searches update	06/23/2021	dh
above searches update	08/13/2021	dh
above searches update	09/10/2021	dh

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
	general interference and searches of claim (PG PUB, US PAT	01/14/2021	dh
	update	06/23/2021	dh
	update	08/13/2021	dh
	update	09/10/2021	dh

/DAO Q HO/ Primary Examiner, Art Unit 2432	
---	--

<i>Index of Claims</i> 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS										
☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47										
CLAIM		DATE								
Final	Original	01/14/2021	06/23/2021	08/13/2021	09/10/2021					
	1	✓	=	=	=					
	2	✓	=	=	=					
	3	✓	=	=	=					
	4	✓	=	=	=					
	5	✓	=	=	=					
	6	✓	=	=	=					
	7	✓	=	=	=					
	8	✓	=	=	=					
	9	✓	=	=	=					
	10	✓	=	=	=					
	11	✓	=	=	=					
	12	✓	=	=	=					
	13	✓	=	=	=					
	14	✓	-	-	-					
	15	✓	=	=	=					
	16	✓	=	=	=					
	17	✓	-	-	-					
	18	✓	=	=	=					
	19	✓	=	=	=					
	20	✓	=	=	=					
	21	✓	=	=	=					
	22	✓	=	=	=					

Doc code: IDS

Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (02-18)

Approved for use through 11/30/2020. OMB 0651-0031

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		16453682
	Filing Date		2019-06-26
	First Named Inventor		John A. Nix
	Art Unit		2432
	Examiner Name	HO, Dao Q.	
	Attorney Docket Number		63121/0184

U.S.PATENTS							Remove
Examiner Initial*	Cite No	Patent Number	Kind Code ¹	Issue Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	
	1						

If you wish to add additional U.S. Patent citation information please click the Add button.

Add

U.S.PATENT APPLICATION PUBLICATIONS							Remove
Examiner Initial*	Cite No	Publication Number	Kind Code ¹	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	
	1						

If you wish to add additional U.S. Published Application citation information please click the Add button.

Add

FOREIGN PATENT DOCUMENTS								Remove
Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ²ⁱ	Kind Code ⁴	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	T ⁵
	1							

If you wish to add additional Foreign Patent Document citation information please click the Add button

Add

NON-PATENT LITERATURE DOCUMENTS				Remove
Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.		T ⁵

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

1	TLS 1.3 Wish List Presentation, IETF 87 (Eric Rescorla), dated August 1, 2013.
---	--

If you wish to add additional non-patent literature document citation information please click the Add button

EXAMINER SIGNATURE

Examiner Signature	/DAO Q HO/	Date Considered	09/10/2021
--------------------	------------	-----------------	------------

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. ² Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). ³ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁴ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁵ Applicant is to place a check mark here if English language translation is attached.

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Albert J. Boardman/	Date (YYYY-MM-DD)	2021-08-25
Name/Print	Albert J. Boardman	Registration Number	70,601

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S124	137	((("NIX") near3 ("John"))).INV.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	OFF	2021/09/10 11:20
S125	52	((("NETWORK-1") near3 ("TECHNOLOGIES") near3 ("INC"))).AS,AANM.	USPAT	OR	OFF	2021/09/10 11:20
S126	171	S124 S125	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:20
S127	37,048	(generat\$4 with signature with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:20
S128	187	S127 and (embed\$4 near universal near integrated near circuit near card)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:20
S129	41	S126 and S128	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:20
S130	211	("10169587" "10621352" "20060021063" "20100093347" "20110158411" "20190313246" "20190332774" "20190356482" "20200162247" "20200162269" "20200226258" "20200280436" "20210126801" "20210184842"	US-PGPUB; USPAT	OR	OFF	2021/09/10 11:29

"20210194681"	"20210211279"				
"20210218560"	"20050058294"				
"20060129848"	"20080044032"				
"20080107083"	"20110035604"				
"20110213959"	"20110268022"				
"20110269422"	"20110269461"				
"20110269472"	"20110270747"				
"20120011362"	"20120087493"				
"20120190354"	"20120263298"				
"20130012168"	"20130122864"				
"20130157673"	"20130165075"				
"20130227646"	"20130283047"				
"20130294602"	"20130340040"				
"20130344864"	"20140003604"				
"20140082358"	"20140082359"				
"20140120874"	"20140140507"				
"20140219447"	"20140235210"				
"20140237101"	"20140308991"				
"20140329502"	"20140357229"				
"20150121495"	"20150281964"				
"20150350881"	"20160057725"				
"20160088096"	"20160127132"				
"20160149903"	"5216715"				
"5642420"	"5664017"	"5852665"			
"5995625"	"6256733"	"6721886"			
"6804357"	"6842628"	"7957532"			
"8352739"	"8452017"	"8577337"			
"8965366"	"9020479"	"9270653"			
"9332129"	"9572016"	"9628981"			
"9674690"	"9794775"	"9800993"			
"9807605"	"9923724"	"10003461"			
"10057059"	"10084768"				
"20010029581"	"20040162472"				
"20040221163"	"20050050323"				
"20060095771"	"20070033403"				
"20070101400"	"20070158439"				
"20080031204"	"20080130879"				
"20080307218"	"20090011320"				
"20090041110"	"20090060197"				
"20090113203"	"20090183541"				
"20090217348"	"20090274306"				
"20100023771"	"20100195833"				
"20100199334"	"20100211779"				
"20100223461"	"20100275028"				
"20110016321"	"20110035584"				
"20110055553"	"20110167272"				
"20110237281"	"20110291803"				
"20110314287"	"20120011360"				
"20120023336"	"20120030461"				
"20120033613"	"20120072732"				
"20120084568"	"20120089568"				
"20120108205"	"20120117635"				
"20120159153"	"20120170451"				

		"20120214444" "20120260086" "20120260090" "20120260095" "20120272064" "20120278490" "20120300932" "20120331287" "20120331292" "20120331298" "20130007442" "20130028184" "20130091556" "20130114810" "20130117824" "20130145151" "20130149996" "20130165073" "20130166915" "20130173747" "20130173926" "20130179962" "20130182586" "20130183932" "20130212637" "20130231087" "20130305345" "20130322621" "20130331063" "20140040628" "20140049653" "20140053241" "20140073375" "20140087790" "20140101444" "20140108801" "20140115335" "20140122878" "20140140509" "20140143534" "20140143826" "20140161113" "20140165155" "20140192976" "20140219448" "20140244994" "20140273913" "20140281483" "20140287725" "20140337937" "20140351403" "20150012743" "20150017910" "20150071139" "20150089214" "20150092590" "20150095514" "20150095648" "20150113275" "20150215126" "20150222604" "20150222619" "20160014112" "20160212617" "20160269386" "20160294829" "20170206532" "20170373845" "20180254897" "20180262329" "20180270059" "5319710" "6038322" "7627760" "7921292" "8127142" "8434139" "8522013" "8526606" "8555067" "8590028" "8713320" "8843179").PN.				
S131	16	("8924715" "9100175" "9118464" "9253643" "9276740" "9300473" "9319223" "9351162" "9408012" "9596078" "9641327" "9698981" "9742562" "9961060" "9998280" "9998281").PN.	US-PGPUB; USPAT	OR	OFF	2021/09/10 11:29
S132	227	S130 S131	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2021/09/10 11:29

			DERWENT; IBM_TDB			
S133	25	S128 and S132	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:29
S134	37,048	(generat\$4 with signature with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:38
S135	187	S134 and (embed\$4 near universal near integrated near circuit near card)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:38
S136	140	S135 AND ((H04W12/06 OR H04W4/60 OR H04L63/0823 OR H04L9/3247 OR H04L63/0853 OR H04L63/08).CPC.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:38
S137	51	((((credit circuit) near card) euicc uicc esim sim) with (nonce random\$2) with signature with (encrypt\$4 generat\$4) with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:41
S138	21	S136 and S137	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 11:41
S139	17,951	((((credit circuit) near card) euicc uicc esim sim) with subscri\$6 with authenticat\$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2021/09/10 12:00

			DERWENT; IBM_TDB			
S140	144	S139 and ((nonce random\$2) with signature with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 12:00
S141	41	S140 and (key near exchange)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 12:00
S142	37	S141 AND ((H04W12/06 OR H04W4/60 OR H04L63/0823 OR H04L9/3247 OR H04L63/0853 OR H04L63/08).CPC.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 12:00
S143	60	S135 and ((nonce random\$2) with signature with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 12:03
S144	36	S143 and (key near exchange)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 12:03
S145	33	S144 AND ((H04W12/06 OR H04W4/60 OR H04L63/0823 OR H04L9/3247 OR H04L63/0853 OR H04L63/08).CPC.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 12:04
S146	37,048	(generat\$4 with signature with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2021/09/10 12:37

			DERWENT; IBM_TDB			
S147	187	S146 and (embed\$4 near universal near integrated near circuit near card)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 12:37
S148	63	(encrypt\$4 with profile with ciphertext\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 12:37
S149	5	S147 and S148	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 12:37
S150	63	(encrypt\$4 with profile with ciphertext\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 13:36
S151	5	S128 and S150	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 13:36
S152	216	(subscript\$4 with (encrypt\$4 near3 profile))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 13:48
S153	156	S139 and S152	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2021/09/10 13:48

			DERWENT; IBM_TDB			
S154	88	S153 and (embed\$4 near universal near integrated near circuit near card)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 13:48
S155	13	S132 and S154	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 13:49
S156	9	S155 and (key near exchange)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 13:50
S157	9	S156 AND ((H04W12/06 OR H04W4/60 OR H04L63/0823 OR H04L9/3247 OR H04L63/0853 OR H04L63/08).CPC.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/09/10 13:50

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S158	819	((((smart credit circuit) near card) euicc uicc esim sim) with subscri\$6 with authenticat\$4).clm.	US-PGPUB; USPAT	OR	ON	2021/09/10 11:30
S159	44	S158 and (key with (random near number)).clm.	US-PGPUB; USPAT	OR	ON	2021/09/10 11:30
S160	11	S159 and (digital near signature).clm.	US-PGPUB; USPAT	OR	ON	2021/09/10 11:30
S161	6,233	(generat\$4 with signature with key).clm.	US-PGPUB; USPAT	OR	ON	2021/09/10 12:00

S162	10	(encrypt\$4 with profile with ciphertext\$2).clm.	US- PGPUB; USPAT	OR	ON	2021/09/10 12:00
S163	5	S161 and S162	US- PGPUB; USPAT	OR	ON	2021/09/10 12:00

9/10/2021 2:13:17 PM

C:\Users\dho1\Documents\EAST\Workspaces\16453682.wsp

Bibliographic Data

Application No: 16/453,682

Foreign Priority claimed: ☐ Yes ☒ No

35 USC 119 (a-d) conditions met: ☐ Yes ☒ No ☐ Met After Allowance

Verified and Acknowledged: /DAO Q HO/

Examiner's Signature

Initials

Title:

Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
06/26/2019	713	2432	63121/0184
RULE			

APPLICANTS

Network-1 Technologies, Inc., New York, NY, UNITED STATES

INVENTORS

John A. Nix, Evanston, IL, UNITED STATES

CONTINUING DATA

This application is a CON of 16110804 08/23/2018 PAT 10382422

16110804 is a CON of 15928848 03/22/2018 PAT 10084768

15928848 is a CON of 14751119 06/25/2015 PAT 9961060

14751119 is a CON of 14099329 12/06/2013 PAT 9100175

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

07/08/2019

**** SMALL ENTITY ****

STATE OR COUNTRY

UNITED STATES

ADDRESS

AMSTER, ROTHSTEIN & EBENSTEIN LLP

90 PARK AVENUE

NEW YORK, NY 10016

UNITED STATES

FILING FEE RECEIVED

\$1,085

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Network-1 Technologies, Inc. Confirmation No.: 2186
Title: Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication
Appl. No.: 16/453,682
Filing Date: June 26, 2019
Examiner: HO, Dao Q.
Art Unit: 2432
Docket No.: 63121/0184
Customer No.: 01912

INFORMATION DISCLOSURE STATEMENT LETTER
UNDER 37 C.F.R. § 1.56

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the duty of disclosure under 37 C.F.R. §1.56, the attention of the Patent and Trademark Office is hereby directed to the information item(s) listed on the attached Form PTO-SB08. Unless otherwise indicated herein, one copy of each item(s) is attached. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the item(s) be made of record therein and appear among the "References Cited" on any patent to issue therefrom.

Reference to Related Patents and Co-Pending Applications

The below chart provides information regarding related issued patents and co-pending applications filed by the present Applicant:

Appl. No. 16/453,682
Information Disclosure Statement Letter

Atty Dkt. No. 63121/0184

Serial No.	Attorney Docket No.	Filing Date	Status
17/304,922	63121/0209	2021-06-28	Application Undergoing Preexam Processing
17/249,242	63121/0202	2021-02-24	Application Dispatched from Preexam
16/879,325	63121/0198	2020-05-20	USP 11,082,218
16/843,107	63121/0197	2020-04-08	Non-Final Rejection mailed 2021-07-02
16/721,153	63121/0195	2019-12-19	Final Office Action mailed 2021-04-15
16/593,561	63121/0190	2019-10-04	Non-Final Rejection mailed 2021-07-06
16/453,682	63121/0184	2019-06-26	RCE Filed concurrently herewith
16/271,455	63121/0181	2019-02-08	USP 10,594,679
16/271,428	63121/0180	2019-02-08	USP 10,523,432
16/201,401	63121/0179	2018-11-27	USP 10,700,856
16/200,118	63121/0178	2018-11-26	USP 10,498,530
16/110,804	63121/0168	2018-08-23	USP 10,382,422
16/036,506	63121/0167	2018-07-16	USP 10,530,575
15/983,957	63121/0163	2018-05-18	USP 10,177,911
15/973,012	63121/0162	2018-05-07	USP 10,250,386
15/972,914	63121/0161	2018-05-07	USP 10,652,017
15/928,848	63121/0160	2018-03-22	USP 10,084,768
15/010,905	63121/0157	2016-01-29	USP 9,641,327
15/680,758	63121/0147	2017-08-18	USP 10,187,206
15/162,292	63121/0146	2016-05-23	USP 10,362,012
14/751,119	63121/0145	2015-06-25	USP 9,961,060
15/457,700	63121/0144	2017-03-13	USP 9,998,281
15/583,968	63121/0143	2017-05-01	USP 10,057,059
15/043,293	63121/0142	2016-02-12	USP 9,998,280
15/642,088	63121/0141	2017-07-05	USP 10,003,461
15/130,146	63121/0140	2016-04-15	USP 9,742,562
15/162,302	63121/0139	2016-05-23	USP 9,698,981
14/789,255	63121/0138	2015-07-01	USP 9,596,078
14/139,419	63121/0137	2013-12-23	USP 9,351,162
14/136,711	63121/0136	2013-12-20	USP 9,300,473
14/099,329	63121/0135	2013-12-06	USP 9,100,175
14/084,141	63121/0134	2013-11-19	USP 9,319,223
14/064,618	63121/0133	2013-10-28	USP 9,118,464
14/055,606	63121/0132	2013-10-16	USP 9,276,740
14/039,401	63121/0131	2013-09-27	USP 9,288,059

Appl. No. 16/453,682
Information Disclosure Statement Letter

Atty Dkt. No. 63121/0184

14/023,181	63121/0130	2013-09-10	USP 9,350,550
AU 2019246774	63121-0199	2019-10-08	Pending
GB 1608573	63121-0151	2014-10-27	Published on 2016-08-03
GB 2100530	63121-0201	2014-10-27	Published on 2021-05-12
CA 2,965,119	63121-0152	2014-10-27	Published on 2015-05-07
HK 17101082.8	63121-0153	2014-10-27	Published on 2017-10-20
EU 14868381.6	63121-0154	2014-12-04	Published on 2017-01-04
CA 2,969,829	63121-0155	2014-12-04	Published on 2015-06-11
HK 17106540.3	63121-0156	2014-12-04	Published on 2018-01-19

The Examiner is invited to review the prosecution history of each of these applications, including the cited references of record and related office actions, as the Examiner deems appropriate.

* * * * *

Since this Supplemental Information Disclosure Statement is being filed more than three months after the U.S. filing date and after the mailing date of a Notice of Allowance, but accompanies a Request for Continued Examination, no certification or fee is required. Please charge any fee deficiency or credit any overpayment to Deposit Account No. 01-1785, under Docket No. 63121/0184.

Respectfully submitted,

/DAO Q HO/

09/10/2021

AMSTER, ROTHSTEIN & EBENSTEIN LLP
Attorneys for Applicant
90 Park Avenue
New York, NY 10016
(212) 336-8000

Dated: August 25, 2021

By: /Albert J. Boardman/
Albert J. Boardman, Reg. No. 70,601

Page 3 of 3

**REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL
(Submitted Only via EFS-Web)**

Application Number	16/453,682	Filing Date	2019-06-26	Docket Number (if applicable)	63121/0184	Art Unit	2432
First Named Inventor	John A. Nix			Examiner Name	HO, Dao Q.		

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.

Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV

SUBMISSION REQUIRED UNDER 37 CFR 1.114

Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

☐ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____☐ Other _____☒ Enclosed☐ Amendment/Reply☒ Information Disclosure Statement (IDS)☐ Affidavit(s)/ Declaration(s)☐ Other _____**MISCELLANEOUS**☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months _____
(Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)☐ Other _____**FEES****The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.**☒ The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to
Deposit Account No 011785**SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED**☒ Patent Practitioner Signature
Applicant Signature

Signature of Registered U.S. Patent Practitioner			
Signature	Albert J. Boardman/	Date (YYYY-MM-DD)	2021-08-25
Name	Albert J. Boardman	Registration Number	70601

This collection of information is required by 37 CFR 1.114. The information is required to obtain or retain a benefit by the public which is a benefit of a patent (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.



**UNITED STATES
PATENT AND TRADEMARK OFFICE**

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #
16/453,682

RECEIPT DATE / TIME
08/25/2021 11:35:53 AM ET

ATTORNEY DOCKET #
63121/0184

Title of Invention

Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

Application Information

APPLICATION TYPE Utility - Nonprovisional Application
under 35 USC 111(a)

PATENT # -

CONFIRMATION # 2186

FILED BY Victoria Gilmore

PATENT CENTER # 60291660

AUTHORIZED BY Albert Boardman

CUSTOMER # 1912

FILING DATE 06/26/2019

CORRESPONDENCE ADDRESS -

FIRST NAMED INVENTOR John A. Nix

Payment Information

PAYMENT METHOD
CARD / 1044

PAYMENT TRANSACTION ID
E202180B37000105

PAYMENT AUTHORIZED BY
Victoria Gilmore

FEE CODE	DESCRIPTION	ITEM PRICE(\$)	QUANTITY	ITEM TOTAL(\$)
2620	REQUEST FOR CONTINUED EXAMINATION (RCE)- 2ND AND SUBSEQUENT REQUEST (SEE 37 CFR 1.114)	1000.00	1	1000.00
			TOTAL AMOUNT:	\$1,000.00

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Network-1 Technologies, Inc. Confirmation No.: 2186
Title: Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication
Appl. No.: 16/453,682
Filing Date: June 26, 2019
Examiner: HO, Dao Q.
Art Unit: 2432
Docket No.: 63121/0184
Customer No.: 01912

INFORMATION DISCLOSURE STATEMENT LETTER
UNDER 37 C.F.R. § 1.56

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the duty of disclosure under 37 C.F.R. §1.56, the attention of the Patent and Trademark Office is hereby directed to the information item(s) listed on the attached Form PTO-SB08. Unless otherwise indicated herein, one copy of each item(s) is attached. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the item(s) be made of record therein and appear among the "References Cited" on any patent to issue therefrom.

Reference to Related Patents and Co-Pending Applications

The below chart provides information regarding related issued patents and co-pending applications filed by the present Applicant:

Serial No.	Attorney Docket No.	Filing Date	Status
17/304,922	63121/0209	2021-06-28	Application Undergoing Preexam Processing
17/249,242	63121/0202	2021-02-24	Application Dispatched from Preexam
16/879,325	63121/0198	2020-05-20	USP 11,082,218
16/843,107	63121/0197	2020-04-08	Non-Final Rejection mailed 2021-07-02
16/721,153	63121/0195	2019-12-19	Final Office Action mailed 2021-04-15
16/593,561	63121/0190	2019-10-04	Non-Final Rejection mailed 2021-07-06
16/453,682	63121/0184	2019-06-26	RCE Filed concurrently herewith
16/271,455	63121/0181	2019-02-08	USP 10,594,679
16/271,428	63121/0180	2019-02-08	USP 10,523,432
16/201,401	63121/0179	2018-11-27	USP 10,700,856
16/200,118	63121/0178	2018-11-26	USP 10,498,530
16/110,804	63121/0168	2018-08-23	USP 10,382,422
16/036,506	63121/0167	2018-07-16	USP 10,530,575
15/983,957	63121/0163	2018-05-18	USP 10,177,911
15/973,012	63121/0162	2018-05-07	USP 10,250,386
15/972,914	63121/0161	2018-05-07	USP 10,652,017
15/928,848	63121/0160	2018-03-22	USP 10,084,768
15/010,905	63121/0157	2016-01-29	USP 9,641,327
15/680,758	63121/0147	2017-08-18	USP 10,187,206
15/162,292	63121/0146	2016-05-23	USP 10,362,012
14/751,119	63121/0145	2015-06-25	USP 9,961,060
15/457,700	63121/0144	2017-03-13	USP 9,998,281
15/583,968	63121/0143	2017-05-01	USP 10,057,059
15/043,293	63121/0142	2016-02-12	USP 9,998,280
15/642,088	63121/0141	2017-07-05	USP 10,003,461
15/130,146	63121/0140	2016-04-15	USP 9,742,562
15/162,302	63121/0139	2016-05-23	USP 9,698,981
14/789,255	63121/0138	2015-07-01	USP 9,596,078
14/139,419	63121/0137	2013-12-23	USP 9,351,162
14/136,711	63121/0136	2013-12-20	USP 9,300,473
14/099,329	63121/0135	2013-12-06	USP 9,100,175
14/084,141	63121/0134	2013-11-19	USP 9,319,223
14/064,618	63121/0133	2013-10-28	USP 9,118,464
14/055,606	63121/0132	2013-10-16	USP 9,276,740
14/039,401	63121/0131	2013-09-27	USP 9,288,059

14/023,181	63121/0130	2013-09-10	USP 9,350,550
AU 2019246774	63121-0199	2019-10-08	Pending
GB 1608573	63121-0151	2014-10-27	Published on 2016-08-03
GB 2100530	63121-0201	2014-10-27	Published on 2021-05-12
CA 2,965,119	63121-0152	2014-10-27	Published on 2015-05-07
HK 17101082.8	63121-0153	2014-10-27	Published on 2017-10-20
EU 14868381.6	63121-0154	2014-12-04	Published on 2017-01-04
CA 2,969,829	63121-0155	2014-12-04	Published on 2015-06-11
HK 17106540.3	63121-0156	2014-12-04	Published on 2018-01-19

The Examiner is invited to review the prosecution history of each of these applications, including the cited references of record and related office actions, as the Examiner deems appropriate.

* * * * *

Since this Supplemental Information Disclosure Statement is being filed more than three months after the U.S. filing date and after the mailing date of a Notice of Allowance, but accompanies a Request for Continued Examination, no certification or fee is required. Please charge any fee deficiency or credit any overpayment to Deposit Account No. 01-1785, under Docket No. 63121/0184.

Respectfully submitted,

AMSTER, ROTHSTEIN & EBENSTEIN LLP
Attorneys for Applicant
90 Park Avenue
New York, NY 10016
(212) 336-8000

Dated: August 25, 2021

By: /Albert J. Boardman/
Albert J. Boardman, Reg. No. 70,601

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

U.S.PATENTS							Remove	
Examiner Initial*	Cite No	Patent Number	Kind Code ¹	Issue Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear		
	1							
If you wish to add additional U.S. Patent citation information please click the Add button.							Add	
U.S.PATENT APPLICATION PUBLICATIONS							Remove	
Examiner Initial*	Cite No	Publication Number	Kind Code ¹	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear		
	1							
If you wish to add additional U.S. Published Application citation information please click the Add button.							Add	
FOREIGN PATENT DOCUMENTS							Remove	
Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ²ⁱ	Kind Code ⁴	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	T ⁵
	1							
If you wish to add additional Foreign Patent Document citation information please click the Add button								Add
NON-PATENT LITERATURE DOCUMENTS								Remove
Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.						T ⁵

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

1	TLS 1.3 Wish List Presentation, IETF 87 (Eric Rescorla), dated August 1, 2013.
---	--

If you wish to add additional non-patent literature document citation information please click the Add button

Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	
--------------------	--	-----------------	--

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. ² Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). ³ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁴ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁵ Applicant is to place a check mark here if English language translation is attached.

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Albert J. Boardman/	Date (YYYY-MM-DD)	2021-08-25
Name/Print	Albert J. Boardman	Registration Number	70,601

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.



ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #
16/453,682

RECEIPT DATE / TIME
08/25/2021 11:35:53 AM ET

ATTORNEY DOCKET #
63121/0184

Title of Invention

Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

Application Information

APPLICATION TYPE Utility - Nonprovisional Application
under 35 USC 111(a)

PATENT # -

CONFIRMATION # 2186

FILED BY Victoria Gilmore

PATENT CENTER # 60291660

FILING DATE 06/26/2019

CUSTOMER # 1912

FIRST NAMED INVENTOR John A. Nix

CORRESPONDENCE ADDRESS -

AUTHORIZED BY Albert Boardman

Documents

TOTAL DOCUMENTS: 4

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
1-RCE.pdf	3	Request for Continued Examination (RCE)	1318 KB
2-TransmittalLetter.pdf	3	Transmittal Letter	126 KB
3-IDS.pdf	4	Information Disclosure Statement (IDS) Form (SB08)	1010 KB
4-NPL1-TLS1-3WishList.pdf	11	Non Patent Literature	36 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
----------	-------------------------

1-RCE.pdf	1D657EB92F64B304FA618F17AD32B5B4A8BB9024D2D97CC80 9DE3DC912BEBBBE010FC56CD64145328DD4B5E4135B3C9FC 311B18A5BE22E17C91F59CC3AD76124
2-TransmittalLetter.pdf	7E6179FD8DC37C10020B2E985CF81C2B77B305250DEB7A518 C42CA7E539BAD3E14E565F23EB75592CF8F8E79865239943E 2432B59033128D38FFB12AEE637DD4
3-IDS.pdf	AF387A285D3D5DA8CD9B61FE098160445BFFB9CCD9012F1B FF4E6C7A5766FAF008375001CEE1E9CBFFE31E6496CAE7219 7BD51200B7AE8441D7472B4B9CBA281
4-NPL1-TLS1-3WishList.pdf	38946D42D07309C837FF7FD5688C57F4B15E514BBE2D468C7 099ED222DB0D4D76E3D8CDA8099E6C4050E5DBB42A33D506 7B83883F3FBB8B17798534A4FC4482E

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

1912 7590 08/20/2021
AMSTER, ROTHSTEIN & EBENSTEIN LLP
90 PARK AVENUE
NEW YORK, NY 10016

EXAMINER

HO, DAO Q

ART UNIT

PAPER NUMBER

2432

DATE MAILED: 08/20/2021

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/453,682	06/26/2019	John A. Nix	63121/0184	2186

TITLE OF INVENTION: Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$600	\$0.00	\$0.00	\$600	11/22/2021

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 1/2 the amount of undiscounted fees, and micro entity fees are 1/2 the amount of small entity fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

1912 7590 08/20/2021
AMSTER, ROTHSTEIN & EBENSTEIN LLP
90 PARK AVENUE
NEW YORK, NY 10016

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

(Typed or printed name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/453,682	06/26/2019	John A. Nix	63121/0184	2186

TITLE OF INVENTION: Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	\$600	\$0.00	\$0.00	\$600	11/22/2021

EXAMINER	ART UNIT	CLASS-SUBCLASS
HO, DAO Q	2432	713-171000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

- ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-09 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

- (1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
- (2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. Fees submitted: ☐ Issue Fee ☐ Publication Fee (if required) ☐ Advance Order - # of Copies _____

4b. Method of Payment: (Please first reapply any previously paid fee shown above)

- ☐ Electronic Payment via EFS-Web ☐ Enclosed check ☐ Non-electronic payment by credit card (Attach form PTO-2038)

☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

5. Change in Entity Status (from status indicated above)

- ☐ Applicant certifying micro entity status. See 37 CFR 1.29
- ☐ Applicant asserting small entity status. See 37 CFR 1.27
- ☐ Applicant changing to regular undiscounted fee status.

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____

Date _____

Typed or printed name _____

Registration No. _____



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
16/453,682	06/26/2019	John A. Nix	63121/0184	2186
1912	7590	08/20/2021	EXAMINER	
AMSTER, ROTHSTEIN & EBENSTEIN LLP			HO, DAO Q	
90 PARK AVENUE			ART UNIT	PAPER NUMBER
NEW YORK, NY 10016			2432	
DATE MAILED: 08/20/2021				

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No. 16/453,682	Applicant(s) Nix, John A.	
	Examiner DAO Q HO	Art Unit 2432	AIA (FITF) Status Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 08/05/2021.
☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on ____.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on ____; the restriction requirement and election have been incorporated into this action.

3. ☒ The allowed claim(s) is/are 1-13, 15-16 and 18-22. As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov**.

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a) ☐ All b) ☐ Some* c) ☐ None of the:

1. ☐ Certified copies of the priority documents have been received.

2. ☐ Certified copies of the priority documents have been received in Application No. ____.

3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: ____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date ____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)	5. ☐ Examiner's Amendment/Comment
2. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date ____.	6. ☒ Examiner's Statement of Reasons for Allowance
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material ____.	7. ☐ Other ____.
4. ☐ Interview Summary (PTO-413), Paper No./Mail Date ____.	

/DAO Q HO/ Primary Examiner, Art Unit 2432	
---	--

Notice of Pre-AIA or AIA Status

The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

DETAILED ACTION

Response to Amendment

This is a reply to the request for Continued Examination (RCE) filed on 08/05/2021, in which Claim(s) 1-13, 15-16 and 18-22 are presented for examination.

Claim(s) 14, 17 are cancelled.

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 08/05/2021 has been entered.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on 07/27/2021 (2) and 08/05/2021 is (are) being considered by the examiner.

Allowable Subject Matter

Claims 1-13, 15-16 and 18-22 are allowed.

The following is an examiner's statement of reasons for allowance:

This communication warrants No Examiner's Reason for Allowance as it was previously stated in the Notice of Allowance filed on 06/29/2021, as such the reasons for allowance are in all probability evident from the record and no statement is deemed necessary (see MPEP 1302.14).

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DAO Q HO whose telephone number is (571)270-5998. The examiner can normally be reached on 7:00am - 5:00pm.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at <http://www.uspto.gov/interviewpractice>.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Nickerson can be reached on (469) 295-9235. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

***/DAO Q HO/
Primary Examiner, Art Unit 2432***

<i>Index of Claims</i> 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS										
☒ Claims renumbered in the same order as presented by applicant ☐ CPA ☒ T.D. ☐ R.1.47										
CLAIM		DATE								
Final	Original	01/14/2021	06/23/2021	08/13/2021						
	1	✓	=	=						
	2	✓	=	=						
	3	✓	=	=						
	4	✓	=	=						
	5	✓	=	=						
	6	✓	=	=						
	7	✓	=	=						
	8	✓	=	=						
	9	✓	=	=						
	10	✓	=	=						
	11	✓	=	=						
	12	✓	=	=						
	13	✓	=	=						
	14	✓	-	-						
	15	✓	=	=						
	16	✓	=	=						
	17	✓	-	-						
	18	✓	=	=						
	19	✓	=	=						
	20	✓	=	=						
	21	✓	=	=						
	22	✓	=	=						

Issue Classification 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.	
	Examiner DAO Q HO	Art Unit 2432	

CPC						
Symbol					Type	Version
H04L	/	63	/	08	F	2013-01-01
H04L	/	9	/	0869	I	2013-01-01
H04L	/	63	/	0428	I	2013-01-01
H04L	/	9	/	0819	I	2013-01-01
H04L	/	9	/	3271	I	2013-01-01
H04W	/	4	/	70	I	2018-02-01
H04W	/	12	/	35	I	2021-01-01
H04L	/	63	/	062	I	2013-01-01
H04W	/	12	/	06	I	2013-01-01
H04B	/	1	/	3816	I	2013-01-01
H04L	/	63	/	0435	I	2013-01-01
H04L	/	63	/	06	I	2013-01-01
H04L	/	63	/	101	I	2013-01-01

CPC Combination Sets					
Symbol				Type	Set
	/		/		

NONE (Assistant Examiner)		Total Claims Allowed: 20	
/DAO Q HO/ Primary Examiner, Art Unit 2432 (Primary Examiner)		13 August 2021 (Date)	O.G. Print Claim(s) 1
			O.G. Print Figure 1

<i>Issue Classification</i> 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.	
	Examiner DAO Q HO	Art Unit 2432	

INTERNATIONAL CLASSIFICATION			
CLAIMED			
H04L	/	29	/ 06
NON-CLAIMED			
	/		/

US ORIGINAL CLASSIFICATION	
CLASS	SUBCLASS

CROSS REFERENCES(S)						
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					

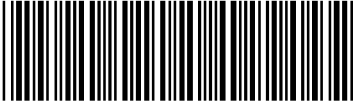
NONE		Total Claims Allowed:	
(Assistant Examiner)	(Date)	20	
/DAO Q HO/ Primary Examiner, Art Unit 2432	13 August 2021	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	1

Issue Classification 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

☒ Claims renumbered in the same order as presented by applicant
 ☐ CPA
 ☒ T.D.
 ☐ R.1.47

CLAIMS															
Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1		10		19										
	2		11		20										
	3		12		21										
	4		13		22										
	5		14												
	6		15												
	7		16												
	8		17												
	9		18												

NONE (Assistant Examiner) _____ (Date) _____		Total Claims Allowed: 20	
/DAO Q HO/ Primary Examiner, Art Unit 2432 (Primary Examiner) _____ (Date) 13 August 2021		O.G. Print Claim(s) 1	O.G. Print Figure 1

<i>Search Notes</i> 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

CPC - Searched*		
Symbol	Date	Examiner
H04L9/0819, 0869, 3247	01/14/2021	dh
H04L63/08, 428, 0853	01/14/2021	dh
H04W12/06	01/14/2021	dh
update	06/23/2021	dh
update	08/13/2021	dh

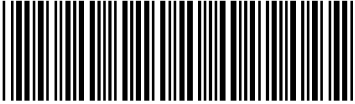
CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
see attached EAST search history	01/14/2021	dh
inventors and assignee search in EAST	01/14/2021	dh
NPL: subscription authenticating using SIM card with key exchange	01/14/2021	dh
CPC searches with delimiter	01/14/2021	dh
above searches update	06/23/2021	dh
above searches update	08/13/2021	dh

/DAO Q HO/ Primary Examiner, Art Unit 2432	
---	--

<i>Search Notes</i> 	Application/Control No. 16/453,682	Applicant(s)/Patent Under Reexamination Nix, John A.
	Examiner DAO Q HO	Art Unit 2432

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
	general interference and searches of claim (PGPUB, USPAT	01/14/2021	dh
	update	06/23/2021	dh
	update	08/13/2021	dh

/DAO Q HO/ Primary Examiner, Art Unit 2432	
---	--

Bibliographic Data

Application No: 16/453,682

Foreign Priority claimed: ☐ Yes ☒ No

35 USC 119 (a-d) conditions met: ☐ Yes ☒ No ☐ Met After Allowance

Verified and Acknowledged: /DAO Q HO/

Examiner's Signature

Initials

Title:

Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
06/26/2019	713	2432	63121/0184
RULE			

APPLICANTS

Network-1 Technologies, Inc., New York, NY, UNITED STATES

INVENTORS

John A. Nix, Evanston, IL, UNITED STATES

CONTINUING DATA

This application is a CON of 16110804 08/23/2018 PAT 10382422

16110804 is a CON of 15928848 03/22/2018 PAT 10084768

15928848 is a CON of 14751119 06/25/2015 PAT 9961060

14751119 is a CON of 14099329 12/06/2013 PAT 9100175

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

07/08/2019

**** SMALL ENTITY ****

STATE OR COUNTRY

UNITED STATES

ADDRESS

AMSTER, ROTHSTEIN & EBENSTEIN LLP

90 PARK AVENUE

NEW YORK, NY 10016

UNITED STATES

FILING FEE RECEIVED

\$1,085

Doc code: IDS

Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (02-18)

Approved for use through 11/30/2020. OMB 0651-0031

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		16453682
	Filing Date		2019-06-26
	First Named Inventor	John A. Nix	
	Art Unit	2432	
	Examiner Name	HO, Dao Q.	
	Attorney Docket Number	63121/0184	

U.S.PATENTS						Remove
Examiner Initial*	Cite No	Patent Number	Kind Code ¹	Issue Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear
	1	10169587	B1	2019-01-01	Nix	
	2	10621352	B2	2020-04-14	Nix	

If you wish to add additional U.S. Patent citation information please click the Add button.

[Add](#)

U.S.PATENT APPLICATION PUBLICATIONS						Remove
Examiner Initial*	Cite No	Publication Number	Kind Code ¹	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear
	1	20060021063	A1	2006-01-26	Hori	
	2	20100093347	A1	2010-04-15	Hahn et al.	
	3	20110158411	A1	2011-06-30	Medvinsky et al.	
	4	20210126801	A1	2021-04-29	Nix	

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

5	20200280436	A1	2020-09-03	Nix	
6	20210218560	A1	2021-07-15	Nix	
7	20210184842	A1	2021-06-17	Nix	
8	20210194681	A1	2021-06-24	Nix	
9	20190356482	A1	2019-11-21	Nix	
10	20200226258	A1	2020-07-16	Nix	
11	20190313246	A1	2019-10-10	Nix	
12	20210211279	A1	2021-07-08	Nix	
13	20200162269	A1	2020-05-21	Nix	
14	20200162247	A1	2020-05-21	Nix	
15	20190332774	A1	2019-10-31	Nix	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

If you wish to add additional U.S. Published Application citation information please click the Add button. [Add](#)

FOREIGN PATENT DOCUMENTS

[Remove](#)

Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ²	Kind Code ⁴	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	T ⁵
	1	2518976	GB	A	2015-04-08	Vodafone IP Licensing Ltd		
	2	2021007235	WO	A1	2021-01-14	Nix		
	3	2021007235	WO	A1	2021-01-14	Nix		
	4	2021087221	WO	A1	2021-05-06	Nix		
	5	2020223319	WO	A1	2020-11-05	Nix		
	6	2020006162	WO	A1	2020-01-02	Nix		
	7	2019246206	WO	A1	2019-12-26	Nix		
	8	2019222319	WO	A1	2019-11-21	Nix		
	9	2019209598	WO	A1	2019-10-31	Nix		

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

10	2019178312	WO	A1	2019-09-19	Nix		
11	2020123959	WO	A1	2020-06-18	Nix		
12	2020061023	WO	A1	2020-03-26	Nix		
13	3110641	CA	A1	2019-10-31	Nix		
14	3110468	CA	A1	2019-11-21	Nix		
15	19803936	DE	A1	1999-08-05	Osram GmbH Siemens AG		
16	3766222	EP	A1	2021-01-20	Nix		

If you wish to add additional Foreign Patent Document citation information please click the Add button

Add

NON-PATENT LITERATURE DOCUMENTS

Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T ⁵
	1	Search Report issued in UK Application No. GB2108534.5 dated July 7, 2021.	

If you wish to add additional non-patent literature document citation information please click the Add button

Add

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

EXAMINER SIGNATURE

Examiner Signature	/DAO Q HO/	Date Considered	08/12/2021
--------------------	------------	-----------------	------------

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. ² Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). ³ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁴ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁵ Applicant is to place a check mark here if English language translation is attached.

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Albert J. Boardman/	Date (YYYY-MM-DD)	2021-08-05
Name/Print	Albert J. Boardman	Registration Number	70,601

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S76	137	((("NIX") near3 ("John"))).INV.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	OFF	2021/08/12 10:04
S77	52	((("NETWORK-1") near3 ("TECHNOLOGIES") near3 ("INC"))).AS,AANM.	USPAT	OR	OFF	2021/08/12 10:04
S78	171	S76 S77	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:04
S79	2	((embedded near universal near integrated near circuit near card) with (nonce random\$2) with signature with (private near key))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:04
S80	1	S78 and S79	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:04
S81	36,750	(generat\$4 with signature with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:05
S82	186	S81 and (embed\$4 near universal near integrated near circuit near card)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:05

S83	139	S82 AND ((H04W12/06 OR H04W4/60 OR H04L63/0823 OR H04L9/3247 OR H04L63/0853 OR H04L63/08).CPC.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:05
S84	361	("10057059" "20080031204" "20090113203" "20130179962" "20140120874" "20150281964" "20160149903" "20180254897" "5216715" "7627760" "8843179" "9332129" "9742562" "9794775" "9807605" "20150113275" "20140161113" "10084768" "20090041110" "20110167272" "20110213959" "20110270747" "20120033613" "20140281483" "20150215126" "5319710" "5642420" "5852665" "5995625" "8434139" "9118464" "9253643" "9300473" "9998280" "20080307218" "20110268022" "20110269422" "20130157673" "20160269386" "20160294829" "20180262329" "20180270059" "8713320" "8965366" "9408012" "9628981" "20120272064" "20040221163" "20070033403" "20070158439" "20090060197" "20110016321" "20130294602" "20130344864" "20140308991" "20150121495" "20160127132" "20160212617" "20170206532" "5664017" "9100175" "9270653" "9319223" "9596078" "20100166167" "20130012168" "20100275028" "20110269461" "20130028184" "20130183932" "20140329502" "8352739" "9020479" "9698981" "20140219447" "20100023771" "20040162472" "20110269472" "20160088096" "20170373845" "6721886" "6842628" "8577337" "8924715" "9641327" "9961060" "20130145151" "20110035584" "20130182586" "20130340040" "20160057725" "6804357" "9276740" "9572016" "9674690" "9800993" "9998281" "20110158411" "20120331287")	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:10

		"20120108205" "10003461" "20090011320" "20140357229" "9351162" "20100211779").PN.				
S85	8	(US-1010828-\$ US-71068403-\$ US- 0169587-\$ US-0621352-\$ US- 16453682-\$ US-6453682-\$).DID.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:13
S87	369	S84 S85	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:14
S88	10	S83 and S87	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:14
S89	63	(encrypt\$4 with profile with ciphertext\$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:36
S90	5	S82 and S89	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 10:36
S91	17,849	(((credit circuit) near card) euicc uicc esim sim) with subscri\$6 with authentica\$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 11:08
S92	532	S91 and (embed\$4 near universal near integrated near circuit near card)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO;	OR	ON	2021/08/12 11:08

			DERWENT; IBM_TDB			
S93	59	S92 and (key near exchange)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 11:08
S94	43	S93 AND ((H04W12/06 OR H04W4/60 OR H04L63/0823 OR H04L9/3247 OR H04L63/0853 OR H04L63/08).CPC.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 11:08
S95	211	("10169587" "10621352" "20060021063" "20100093347" "20110158411" "20190313246" "20190332774" "20190356482" "20200162247" "20200162269" "20200226258" "20200280436" "20210126801" "20210184842" "20210194681" "20210211279" "20210218560" "20050058294" "20060129848" "20080044032" "20080107083" "20110035604" "20110213959" "20110268022" "20110269422" "20110269461" "20110269472" "20110270747" "20120011362" "20120087493" "20120190354" "20120263298" "20130012168" "20130122864" "20130157673" "20130165075" "20130227646" "20130283047" "20130294602" "20130340040" "20130344864" "20140003604" "20140082358" "20140082359" "20140120874" "20140140507" "20140219447" "20140235210" "20140237101" "20140308991" "20140329502" "20140357229" "20150121495" "20150281964" "20150350881" "20160057725" "20160088096" "20160127132" "20160149903" "5216715" "5642420" "5664017" "5852665" "5995625" "6256733" "6721886" "6804357" "6842628" "7957532" "8352739" "8452017" "8577337" "8965366" "9020479" "9270653"	US-PGPUB; USPAT	OR	OFF	2021/08/12 12:16

"9332129"	"9572016"	"9628981"			
"9674690"	"9794775"	"9800993"			
"9807605"	"9923724"	"10003461"			
"10057059"	"10084768"				
"20010029581"	"20040162472"				
"20040221163"	"20050050323"				
"20060095771"	"20070033403"				
"20070101400"	"20070158439"				
"20080031204"	"20080130879"				
"20080307218"	"20090011320"				
"20090041110"	"20090060197"				
"20090113203"	"20090183541"				
"20090217348"	"20090274306"				
"20100023771"	"20100195833"				
"20100199334"	"20100211779"				
"20100223461"	"20100275028"				
"20110016321"	"20110035584"				
"20110055553"	"20110167272"				
"20110237281"	"20110291803"				
"20110314287"	"20120011360"				
"20120023336"	"20120030461"				
"20120033613"	"20120072732"				
"20120084568"	"20120089568"				
"20120108205"	"20120117635"				
"20120159153"	"20120170451"				
"20120214444"	"20120260086"				
"20120260090"	"20120260095"				
"20120272064"	"20120278490"				
"20120300932"	"20120331287"				
"20120331292"	"20120331298"				
"20130007442"	"20130028184"				
"20130091556"	"20130114810"				
"20130117824"	"20130145151"				
"20130149996"	"20130165073"				
"20130166915"	"20130173747"				
"20130173926"	"20130179962"				
"20130182586"	"20130183932"				
"20130212637"	"20130231087"				
"20130305345"	"20130322621"				
"20130331063"	"20140040628"				
"20140049653"	"20140053241"				
"20140073375"	"20140087790"				
"20140101444"	"20140108801"				
"20140115335"	"20140122878"				
"20140140509"	"20140143534"				
"20140143826"	"20140161113"				
"20140165155"	"20140192976"				
"20140219448"	"20140244994"				
"20140273913"	"20140281483"				
"20140287725"	"20140337937"				
"20140351403"	"20150012743"				
"20150017910"	"20150071139"				
"20150089214"	"20150092590"				

		"20150095514" "20150095648" "20150113275" "20150215126" "20150222604" "20150222619" "20160014112" "20160212617" "20160269386" "20160294829" "20170206532" "20170373845" "20180254897" "20180262329" "20180270059" "5319710" "6038322" "7627760" "7921292" "8127142" "8434139" "8522013" "8526606" "8555067" "8590028" "8713320" "8843179").PN.				
S96	16	("8924715" "9100175" "9118464" "9253643" "9276740" "9300473" "9319223" "9351162" "9408012" "9596078" "9641327" "9698981" "9742562" "9961060" "9998280" "9998281").PN.	US-PGPUB; USPAT	OR	OFF	2021/08/12 12:16
S97	227	S96 S95	US-PGPUB; USPAT	OR	ON	2021/08/12 12:18
S98	63	S91 and S97	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 12:19
S99	20	S98 and (key with (random near number))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 12:19
S100	6	S99 and (profile near key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/12 12:20
S106	18,281	(((smart credit circuit) near card) euicc uicc esim sim) with subscri\$6 with authenticat\$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/13 07:21
S107	881	S106 and (key with (random near number))	US-PGPUB; USPAT;	OR	ON	2021/08/13 07:21

			USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB			
S108	147	S106 and ((nonce random\$2) with signature with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/13 07:21
S109	147	S107 and (digital near signature)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/13 07:22
S110	65	S109 and (key near exchange)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/13 07:22
S111	60	S110 AND ((H04W12/06 OR H04W4/60 OR H04L63/0823 OR H04L9/3247 OR H04L63/0853 OR H04L63/08).CPC.)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/13 07:22
S116	36,750	(generat\$4 with signature with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/13 09:50
S117	186	S116 and (embed\$4 near universal near integrated near circuit near card)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/13 09:50
S118	139	S117 AND ((H04W12/06 OR H04W4/60 OR H04L63/0823 OR	US-PGPUB; USPAT; USOCR;	OR	ON	2021/08/13 09:50

		H04L9/3247 OR H04L63/0853 OR H04L63/08).CPC.)	FPRS; EPO; JPO; DERWENT; IBM_TDB			
S119	211	("10169587" "10621352" "20060021063" "20100093347" "20110158411" "20190313246" "20190332774" "20190356482" "20200162247" "20200162269" "20200226258" "20200280436" "20210126801" "20210184842" "20210194681" "20210211279" "20210218560" "20050058294" "20060129848" "20080044032" "20080107083" "20110035604" "20110213959" "20110268022" "20110269422" "20110269461" "20110269472" "20110270747" "20120011362" "20120087493" "20120190354" "20120263298" "20130012168" "20130122864" "20130157673" "20130165075" "20130227646" "20130283047" "20130294602" "20130340040" "20130344864" "20140003604" "20140082358" "20140082359" "20140120874" "20140140507" "20140219447" "20140235210" "20140237101" "20140308991" "20140329502" "20140357229" "20150121495" "20150281964" "20150350881" "20160057725" "20160088096" "20160127132" "20160149903" "5216715" "5642420" "5664017" "5852665" "5995625" "6256733" "6721886" "6804357" "6842628" "7957532" "8352739" "8452017" "8577337" "8965366" "9020479" "9270653" "9332129" "9572016" "9628981" "9674690" "9794775" "9800993" "9807605" "9923724" "10003461" "10057059" "10084768" "20010029581" "20040162472" "20040221163" "20050050323" "20060095771" "20070033403" "20070101400" "20070158439" "20080031204" "20080130879" "20080307218" "20090011320" "20090041110" "20090060197" "20090113203" "20090183541" "20090217348" "20090274306")	US-PGPUB; USPAT	OR	OFF	2021/08/13 09:55

"20100023771"	"20100195833"
"20100199334"	"20100211779"
"20100223461"	"20100275028"
"20110016321"	"20110035584"
"20110055553"	"20110167272"
"20110237281"	"20110291803"
"20110314287"	"20120011360"
"20120023336"	"20120030461"
"20120033613"	"20120072732"
"20120084568"	"20120089568"
"20120108205"	"20120117635"
"20120159153"	"20120170451"
"20120214444"	"20120260086"
"20120260090"	"20120260095"
"20120272064"	"20120278490"
"20120300932"	"20120331287"
"20120331292"	"20120331298"
"20130007442"	"20130028184"
"20130091556"	"20130114810"
"20130117824"	"20130145151"
"20130149996"	"20130165073"
"20130166915"	"20130173747"
"20130173926"	"20130179962"
"20130182586"	"20130183932"
"20130212637"	"20130231087"
"20130305345"	"20130322621"
"20130331063"	"20140040628"
"20140049653"	"20140053241"
"20140073375"	"20140087790"
"20140101444"	"20140108801"
"20140115335"	"20140122878"
"20140140509"	"20140143534"
"20140143826"	"20140161113"
"20140165155"	"20140192976"
"20140219448"	"20140244994"
"20140273913"	"20140281483"
"20140287725"	"20140337937"
"20140351403"	"20150012743"
"20150017910"	"20150071139"
"20150089214"	"20150092590"
"20150095514"	"20150095648"
"20150113275"	"20150215126"
"20150222604"	"20150222619"
"20160014112"	"20160212617"
"20160269386"	"20160294829"
"20170206532"	"20170373845"
"20180254897"	"20180262329"
"20180270059"	"5319710"
"6038322"	"7627760"
"8127142"	"8434139"
"8526606"	"8555067"
"8713320"	"8843179").PN.

S120	16	("8924715" "9100175" "9118464" "9253643" "9276740" "9300473" "9319223" "9351162" "9408012" "9596078" "9641327" "9698981" "9742562" "9961060" "9998280" "9998281").PN.	US-PGPUB; USPAT	OR	OFF	2021/08/13 09:55
S121	227	S120 S119	US-PGPUB; USPAT	OR	ON	2021/08/13 09:55
S122	20	S118 and S121	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/13 09:55
S123	60	S117 and ((nonce random\$2) with signature with key)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2021/08/13 09:55

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S101	6,162	(generat\$4 with signature with key).clm.	US-PGPUB; USPAT	OR	ON	2021/08/12 10:15
S102	10	(encrypt\$4 with profile with ciphertext\$2).clm.	US-PGPUB; USPAT	OR	ON	2021/08/12 10:15
S103	5	S101 and S102	US-PGPUB; USPAT	OR	ON	2021/08/12 10:15
S104	757	(((((credit circuit) near card) euicc uicc esim sim) with subscri\$6 with authenticat\$4).clm.	US-PGPUB; USPAT	OR	ON	2021/08/12 10:35
S105	12	S104 and ((nonce random\$2) with signature with key).clm.	US-PGPUB; USPAT	OR	ON	2021/08/12 10:35
S112	813	(((((smart credit circuit) near card) euicc uicc esim sim) with subscri\$6 with authenticat\$4).clm.	US-PGPUB; USPAT	OR	ON	2021/08/13 07:21
S113	12	S112 and ((nonce random\$2) with signature with key).clm.	US-PGPUB; USPAT	OR	ON	2021/08/13 07:21

S114	43	S112 and (key with (random near number)).clm.	US-PGPUB; USPAT	OR	ON	2021/08/13 07:22
S115	11	S114 and (digital near signature).clm.	US-PGPUB; USPAT	OR	ON	2021/08/13 07:22

8/13/2021 10:57:57 AM

C:\Users\dho1\Documents\EAST\Workspaces\16453682.wsp

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Network-1 Technologies, Inc. Confirmation No.: 2186
Title: Embedded Universal Integrated Circuit Card Supporting Two-Factor Authentication
Appl. No.: 16/453,682
Filing Date: June 26, 2019
Examiner: HO, Dao Q.
Art Unit: 2432
Docket No.: 63121/0184
Customer No.: 01912

INFORMATION DISCLOSURE STATEMENT LETTER
UNDER 37 C.F.R. § 1.56

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the duty of disclosure under 37 C.F.R. §1.56, the attention of the Patent and Trademark Office is hereby directed to the information item(s) listed on the attached Form PTO-SB08. Unless otherwise indicated herein, one copy of each item(s) is attached. It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the item(s) be made of record therein and appear among the "References Cited" on any patent to issue therefrom.

Reference to Related Patents and Co-Pending Applications

The below chart provides information regarding related issued patents and co-pending applications filed by the present Applicant:

Appl. No. 16/453,682
Information Disclosure Statement Letter

Atty Dkt. No. 63121/0184

Serial No.	Attorney Docket No.	Filing Date	Status
17/304,922	63121/0209	2021-06-28	Application Undergoing Preexam Processing
17/249,242	63121/0202	2021-02-24	Application Dispatched from Preexam
16/879,325	63121/0198	2020-05-20	Application to issue as USP 11,082,218 on 2021-08-03
16/843,107	63121/0197	2020-04-08	Non-Final Rejection mailed 2021-07-02
16/721,153	63121/0195	2019-12-19	Final Office Action mailed 2021-04-15
16/593,561	63121/0190	2019-10-04	Non-Final Rejection mailed 2021-07-06
16/453,682	63121/0184	2019-06-26	Notice of Allowance mailed 2021-06-29
16/271,455	63121/0181	2019-02-08	USP 10,594,679
16/271,428	63121/0180	2019-02-08	USP 10,523,432
16/201,401	63121/0179	2018-11-27	USP 10,700,856
16/200,118	63121/0178	2018-11-26	USP 10,498,530
16/110,804	63121/0168	2018-08-23	USP 10,382,422
16/036,506	63121/0167	2018-07-16	USP 10,530,575
15/983,957	63121/0163	2018-05-18	USP 10,177,911
15/973,012	63121/0162	2018-05-07	USP 10,250,386
15/972,914	63121/0161	2018-05-07	USP 10,652,017
15/928,848	63121/0160	2018-03-22	USP 10,084,768
15/010,905	63121/0157	2016-01-29	USP 9,641,327
15/680,758	63121/0147	2017-08-18	USP 10,187,206
15/162,292	63121/0146	2016-05-23	USP 10,362,012
14/751,119	63121/0145	2015-06-25	USP 9,961,060
15/457,700	63121/0144	2017-03-13	USP 9,998,281
15/583,968	63121/0143	2017-05-01	USP 10,057,059
15/043,293	63121/0142	2016-02-12	USP 9,998,280
15/642,088	63121/0141	2017-07-05	USP 10,003,461
15/130,146	63121/0140	2016-04-15	USP 9,742,562
15/162,302	63121/0139	2016-05-23	USP 9,698,981
14/789,255	63121/0138	2015-07-01	USP 9,596,078
14/139,419	63121/0137	2013-12-23	USP 9,351,162
14/136,711	63121/0136	2013-12-20	USP 9,300,473
14/099,329	63121/0135	2013-12-06	USP 9,100,175
14/084,141	63121/0134	2013-11-19	USP 9,319,223
14/064,618	63121/0133	2013-10-28	USP 9,118,464
14/055,606	63121/0132	2013-10-16	USP 9,276,740
14/039,401	63121/0131	2013-09-27	USP 9,288,059
14/023,181	63121/0130	2013-09-10	USP 9,350,550

Appl. No. 16/453,682
Information Disclosure Statement Letter

Atty Dkt. No. 63121/0184

AU 2019246774	63121-0199	2019-10-08	Pending
GB 1608573	63121-0151	2014-10-27	Published on 2016-08-03
GB 2100530	63121-0201	2014-10-27	Published on 2021-05-12
CA 2,965,119	63121-0152	2014-10-27	Published on 2015-05-07
HK 17101082.8	63121-0153	2014-10-27	Published on 2017-10-20
EU 14868381.6	63121-0154	2014-12-04	Published on 2017-01-04
CA 2,969,829	63121-0155	2014-12-04	Published on 2015-06-11
HK 17106540.3	63121-0156	2014-12-04	Published on 2018-01-19

The Examiner is invited to review the prosecution history of each of these applications, including the cited references of record and related office actions, as the Examiner deems appropriate.

* * * * *

This Information Disclosure Statement is being filed more than three months after the U.S. filing date and after the mailing date of a Notice of Allowance, but before payment of the Issue Fee. Applicant(s) hereby petition(s) that the Information Disclosure Statement be considered. Accompanying this statement is the amount of \$130.00 in payment of the fee under 37 CFR §1.17(p). Please charge any fee deficiency or credit any overpayment to Deposit Account No. 01-1785, under Docket No. 63121/0184.

/DAO Q HO/

08/12/2021

Page 3 of 4

Appl. No. 16/453,682
Information Disclosure Statement Letter

Atty Dkt. No. 63121/0184

Respectfully submitted,

AMSTER, ROTHSTEIN & EBENSTEIN LLP
Attorneys for Applicant
90 Park Avenue
New York, NY 10016
(212) 336-8000

Dated: July 27, 2021

By: /Albert J. Boardman/
Albert J. Boardman
Reg. No. 70,601

Doc code: IDS

Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (02-18)

Approved for use through 11/30/2020. OMB 0651-0031

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number		16453682
	Filing Date		2019-06-26
	First Named Inventor	John A. Nix	
	Art Unit	2432	
	Examiner Name	HO, Dao Q.	
	Attorney Docket Number	63121/0184	

U.S.PATENTS						Remove
Examiner Initial*	Cite No	Patent Number	Kind Code ¹	Issue Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear
	1	10169587	B1	2019-01-01	Nix	
	2	10621352	B2	2020-04-14	Nix	

If you wish to add additional U.S. Patent citation information please click the Add button.

Add

U.S.PATENT APPLICATION PUBLICATIONS						Remove
Examiner Initial*	Cite No	Publication Number	Kind Code ¹	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear
	1	20060021063	A1	2006-01-26	Hori	
	2	20100093347	A1	2010-04-15	Hahn et al.	
	3	20110158411	A1	2011-06-30	Medvinsky et al.	
	4	20210126801	A1	2021-04-29	Nix	

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

5	20200280436	A1	2020-09-03	Nix	
6	20210218560	A1	2021-07-15	Nix	
7	20210184842	A1	2021-06-17	Nix	
8	20210194681	A1	2021-06-24	Nix	
9	20190356482	A1	2019-11-21	Nix	
10	20200226258	A1	2020-07-16	Nix	
11	20190313246	A1	2019-10-10	Nix	
12	20210211279	A1	2021-07-08	Nix	
13	20200162269	A1	2020-05-21	Nix	
14	20200162247	A1	2020-05-21	Nix	
15	20190332774	A1	2019-10-31	Nix	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

If you wish to add additional U.S. Published Application citation information please click the Add button. [Add](#)

FOREIGN PATENT DOCUMENTS

[Remove](#)

Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ²	Kind Code ⁴	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	T ⁵
	1	2518976	GB	A	2015-04-08	Vodafone IP Licensing Ltd		
	2	2021007235	WO	A1	2021-01-14	Nix		
	3	2021007235	WO	A1	2021-01-14	Nix		
	4	2021087221	WO	A1	2021-05-06	Nix		
	5	2020223319	WO	A1	2020-11-05	Nix		
	6	2020006162	WO	A1	2020-01-02	Nix		
	7	2019246206	WO	A1	2019-12-26	Nix		
	8	2019222319	WO	A1	2019-11-21	Nix		
	9	2019209598	WO	A1	2019-10-31	Nix		

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

10	2019178312	WO	A1	2019-09-19	Nix		
11	2020123959	WO	A1	2020-06-18	Nix		
12	2020061023	WO	A1	2020-03-26	Nix		
13	3110641	CA	A1	2019-10-31	Nix		
14	3110468	CA	A1	2019-11-21	Nix		
15	19803936	DE	A1	1999-08-05	Osram GmbH Siemens AG		
16	3766222	EP	A1	2021-01-20	Nix		

If you wish to add additional Foreign Patent Document citation information please click the Add button

Add

NON-PATENT LITERATURE DOCUMENTS

Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T ⁵
	1	Search Report issued in UK Application No. GB2108534.5 dated July 7, 2021.	

If you wish to add additional non-patent literature document citation information please click the Add button

Add

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

EXAMINER SIGNATURE

Examiner Signature	/DAO Q HO/	Date Considered	08/12/2021
--------------------	------------	-----------------	------------

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. ² Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). ³ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁴ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁵ Applicant is to place a check mark here if English language translation is attached.

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☒ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

☒ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Albert J. Boardman/	Date (YYYY-MM-DD)	2021-07-27
Name/Print	Albert J. Boardman	Registration Number	70,601

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

U.S.PATENTS

Remove

Examiner Initial*	Cite No	Patent Number	Kind Code ¹	Issue Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear
	1	10169587	B1	2019-01-01	Nix	
	2	10621352	B2	2020-04-14	Nix	

If you wish to add additional U.S. Patent citation information please click the Add button.

Add

U.S.PATENT APPLICATION PUBLICATIONS

Remove

Examiner Initial*	Cite No	Publication Number	Kind Code ¹	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear
	1	20060021063	A1	2006-01-26	Hori	
	2	20100093347	A1	2010-04-15	Hahn et al.	
	3	20110158411	A1	2011-06-30	Medvinsky et al.	
	4	20210126801	A1	2021-04-29	Nix	

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

5	20200280436	A1	2020-09-03	Nix	
6	20210218560	A1	2021-07-15	Nix	
7	20210184842	A1	2021-06-17	Nix	
8	20210194681	A1	2021-06-24	Nix	
9	20190356482	A1	2019-11-21	Nix	
10	20200226258	A1	2020-07-16	Nix	
11	20190313246	A1	2019-10-10	Nix	
12	20210211279	A1	2021-07-08	Nix	
13	20200162269	A1	2020-05-21	Nix	
14	20200162247	A1	2020-05-21	Nix	
15	20190332774	A1	2019-10-31	Nix	

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

If you wish to add additional U.S. Published Application citation information please click the Add button. [Add](#)

FOREIGN PATENT DOCUMENTS

[Remove](#)

Examiner Initial*	Cite No	Foreign Document Number ³	Country Code ² i	Kind Code ⁴	Publication Date	Name of Patentee or Applicant of cited Document	Pages, Columns, Lines where Relevant Passages or Relevant Figures Appear	T ⁵
	1	2518976	GB	A	2015-04-08	Vodafone IP Licensing Ltd		
	2	2021007235	WO	A1	2021-01-14	Nix		
	3	2021007235	WO	A1	2021-01-14	Nix		
	4	2021087221	WO	A1	2021-05-06	Nix		
	5	2020223319	WO	A1	2020-11-05	Nix		
	6	2020006162	WO	A1	2020-01-02	Nix		
	7	2019246206	WO	A1	2019-12-26	Nix		
	8	2019222319	WO	A1	2019-11-21	Nix		
	9	2019209598	WO	A1	2019-10-31	Nix		

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

10	2019178312	WO	A1	2019-09-19	Nix		
11	2020123959	WO	A1	2020-06-18	Nix		
12	2020061023	WO	A1	2020-03-26	Nix		
13	3110641	CA	A1	2019-10-31	Nix		
14	3110468	CA	A1	2019-11-21	Nix		
15	19803936	DE	A1	1999-08-05	Osram GmbH Siemens AG		
16	3766222	EP	A1	2021-01-20	Nix		

If you wish to add additional Foreign Patent Document citation information please click the Add button

Add

NON-PATENT LITERATURE DOCUMENTS

Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T ⁵
	1	Search Report issued in UK Application No. GB2108534.5 dated July 7, 2021.	

If you wish to add additional non-patent literature document citation information please click the Add button

Add

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

EXAMINER SIGNATURE

Examiner Signature		Date Considered	
--------------------	--	-----------------	--

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. ² Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). ³ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁴ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁵ Applicant is to place a check mark here if English language translation is attached.

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**
(Not for submission under 37 CFR 1.99)

Application Number	16453682
Filing Date	2019-06-26
First Named Inventor	John A. Nix
Art Unit	2432
Examiner Name	HO, Dao Q.
Attorney Docket Number	63121/0184

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Albert J. Boardman/	Date (YYYY-MM-DD)	2021-08-05
Name/Print	Albert J. Boardman	Registration Number	70,601

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.



(51) International Patent Classification:

H04L 29/06 (2006.01) H04W 12/04 (2009.01)
H04L 29/08 (2006.01)

(21) International Application Number:

PCT/US2019/022184

(22) International Filing Date:

14 March 2019 (14.03.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/644,195 16 March 2018 (16.03.2018) US

(71) Applicant: IOT AND M2M TECHNOLOGIES, LLC
[US/US]; 807 Davis Street, Unit 2207, Evanston, IL
60201-7104 (US).

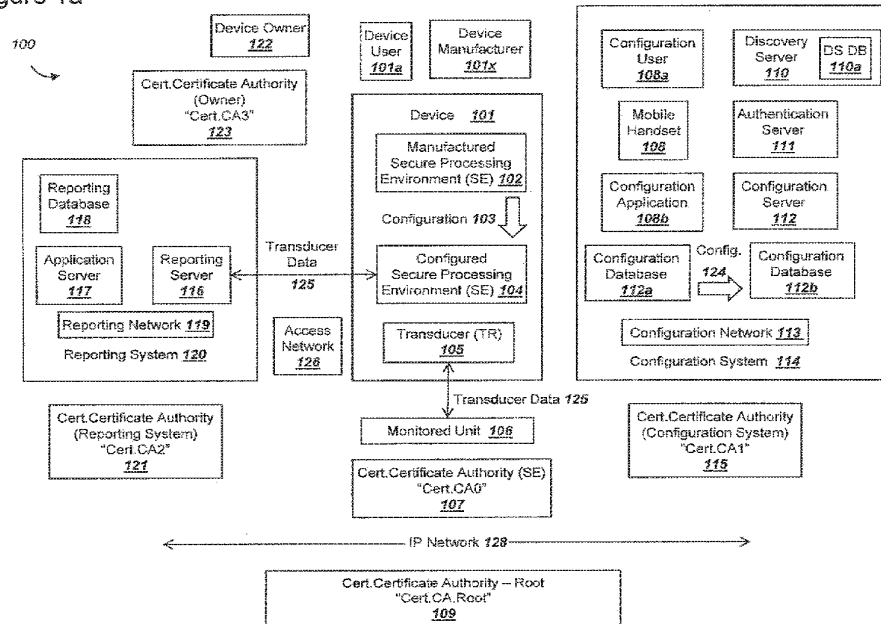
(72) Inventor: NIX, John, A.; c/o IoT And M2M Technologies, LLC, 807 Davis Street, Unit 2207, Evanston, IL 60201-7104 (US).

(74) Agent: THOMSON, Kirsten, L.; McDonnell Boehnen Hulbert & Berghoff LLP, 300 South Wacker Drive, Chicago, IL 60606 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

(54) Title: CONFIGURATION SYSTEMS AND METHODS FOR SECURE OPERATION OF NETWORKED TRANSDUCERS

Figure 1a



(57) **Abstract:** A device can include an internal secure processing environment (SE) and communicate with a configuration system. The SE may utilize a near field communications (NFC) radio. A mobile handset can connect with the SE in the device using NFC. The mobile handset can communicate with the configuration system and receive configuration data and a software package for the device. The configuration system can derive a PKI key pair and send the derived public key to the device via the mobile handset. The SE and the configuration system can mutually derive an encryption key using the derived PKI key pair. The configuration data can be transmitted over the NFC radio, and the mobile handset can establish a Wi-Fi access point. The software package can be encrypted using the encryption key and transmitted to the device over the established Wi-Fi access point, thereby completing a configuration step for the device.

[Continued on next page]

SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

CONFIGURATION SYSTEMS AND METHODS FOR SECURE OPERATION OF NETWORKED TRANSDUCERS

CROSS-REFERENCE TO RELATED APPLICATIONS

5 This international PCT application claims the benefit of the filing date of U.S. Provisional Patent Application Serial No. 62/644,195, filed March 16, 2018, which is hereby incorporated by reference in its entirety.

BACKGROUND

Technical Field

10 The present systems and methods relate to configuration and operation of networked transducers, and more particularly, to systems and methods for securely configuring a secure processing environment within a transducer device, where the transducer device can communicate with a configuration system, a reporting system, and a mobile handset in order to complete configuration.

Description of Related Art

 The ability to connect transducers such as sensors and actuators with a network is a growing field with many economical applications. As the costs for both electronic hardware and bandwidth continues to decrease, the use of networked transducers is expected to continue increasing over the coming decades. Connecting transducers to a
20 network can be referred to as “machine-to-machine (M2M) communications” or “the Internet of Things (IoT).” Among many potential benefits, IoT technologies allow automated monitoring and/or control of assets, equipment, personnel, or a physical location where manual monitoring may not be economical. Many applications for the “Internet of Things” significantly reduce costs by using automated monitoring instead of manual
25 techniques. Prominent examples of IoT applications today include monitoring and control for building heating/air-conditioning, automobiles, alarm systems, and tracking devices. Fast growing markets for IoT applications today include health applications such as the remote monitoring of a person’s fitness activity, heartbeat, or glucose levels, monitoring of industrial equipment deployed remotely in the field, and also security systems.

30 IoT communications can provide remote control over actuators that may be connected to a device, such as turning on or off a power switch, locking or unlocking a door, adjusting a speed of a motor, or similar remote control. A decision to change or adjust an actuator associated with a connected device can utilize one or a series of sensor

measurements. As one example, an IoT device for a truck can periodically report engine status to a remote server, and if the engine is operating outside specifications such as being too hot, including potentially an “alarm” condition, then temperature and an alarm code can be reported to a central server for the IoT device. The server can subsequently instruct the driver and/or a specified mechanic to investigate the engine for potential mechanical malfunctions or other causes. The previous example is just one of many possible applications for IoT technology.

Many IoT applications can leverage wireless networking technologies, in addition to wired technologies such as Ethernet. Wireless technologies such as wireless local area networks and wireless wide area networks have proliferated around the world over the past 20 years, and usage of these wireless networks is also expected to continue to grow. Wireless local area network (LAN) technologies include Wi-Fi and wireless wide area network (WAN) technologies include 3rd Generation Partnership Project’s (3GPP) 3rd Generation (3G) Universal Mobile Telecommunications System (UMTS) and 4th Generation (4G) Long-term Evolution (LTE), LTE Advanced, and Narrow-band Internet of Things (NB-IoT). The many options to connect a transducer device to a network creates opportunities for new products and services, but also creates several classes of problems that need to be solved. Many of these problems or needs in the art pertain to enabling users to securely and efficiently configure the transducer devices. A need exists in the art to allow a user to securely upload to the device a set of network access credentials for the wireless network.

A need exists in the art to support the configuration of transducer devices for installation and operation, including deployment in the field or with monitored units by a user or a technician. A manufactured transducer device can include (i) a secure processing environment which could record a “root of trust” or secret key for the device as well as certificates and cryptographic parameters, and (ii) installed firmware and software. The secure processing environment is often by design isolated in order to enhance security, but device users may need to update the secret key, certificates, and cryptographic parameters. The isolation creates problems for users or device owners to readily access and update the secure processing environment and software for the device. A need exists in the art to allow users or device owners to efficiently change or update for a device both (i) the secret key, certificates, and cryptographic parameters, and (ii) installed firmware and software. A need exists in the art for the users to also efficiently load or transfer network access credentials for a wireless network, in order for the device to obtain network connectivity. A need exists in the art for a configuration system and/or a reporting system to securely record values and identities pertaining to operation of the device, such as a list of

transducers attached, the identity of a monitored unit for the device, as well as the physical location of the device.

Devices with an ARM™-based processor may include a secure processing environment for security operations such as recording keys and certificates. Other processor manufacturers have related security architectures, such as a secure enclave from Apple™ or Intel™. These exemplary processors can be included in a wide variety of devices for machine-to-machine communications, or also applications of “The Internet of Things”. Manufacturers of networked devices with transducers may also sell products with secure elements to increase the secure operation of devices, networks, and servers, where secure operation can comprise confident authentication of users and device, authorization, data confidentiality, and data integrity. Manufacturers of devices may provision a default set of cryptographic values for secure processing environments, secure enclaves, and secure elements, but a need exists in the art to allow device users to update the cryptographic values in order to support the user’s application or environment. In addition, the device and cryptographic values such as keys may have physically been in the hands, possession, or control of third parties before delivery of the device to the user, with the result that the cryptographic values such as keys were also in the hands, possession, or control of the third parties and thus security for a device user could be reduced. A need exists in the art to support a user updating cryptographic values for a secure environment after receipt of the device from a third party.

With conventional technology for configuration and installation of networked devices, a frequent challenge can be that that public key infrastructure (PKI) keys and parameters for default or manufactured configuration of a device may not match the requirements for a configuration system or a reporting system. In other words, the “root of trust” installed by manufacturers of devices and secure enclaves within those devices may have certificates and cryptographic parameters that are not compatible with an operating environment in which the device will be deployed. In other words, the operating environment may use or require (i) different elliptic curve names, (ii) support for different algorithms (ECC, RSA, SIDH, etc.), (iii) more bits of security in the device secret key, etc., than received with a manufactured device. A need exists in the art such that device private keys, cryptographic algorithms, and cryptographic parameters for a user’s application can be securely updated after device manufacturing. A need exists in the art to support the update through a highly automated configuration step.

In addition, devices designed for machine-to-machine communications, or the “Internet of Things”, can frequently be shipped to an end user in a state that is not fully configured for the user. The lack of a full configuration could include incomplete

information for any of the following: (i) current, running firmware or configuration files for the device, (ii) network access credentials, (iii) credentials for access to remote servers, and (iv) proper configuration to support transducers that are actually connected with a device. As one example, hospitals, health clinics, or doctor's offices can receive health monitoring equipment designed to be networked, but without both (i) the networking configured for the equipment, and (ii) updated or patched firmware for the equipment. A need exists in the art to allow general users (e.g. not requiring expert IT skills) to securely configure equipment designed to be networked, including securely updating the device's firmware and credentials for both accessing a network and remote servers.

Many other examples exist as well for needs in the art to support efficient yet secure device configuration by users, and the above are examples are just a few and intended to be illustrative instead of limiting.

SUMMARY

Methods and systems are provided for configuration of networked transducers in order to support secure operation of the networked transducers. The methods and systems can support secure operation of devices for “The Internet of Things (IoT)” which is also known as “Machine to Machine (M2M)” communications. An objective of the invention is to address the challenges noted above for securing the deployment and operation of devices that have radios and transducers, where the devices connect to servers using networks based on Internet protocols.

A device can have a transducer for monitoring a monitored unit, and the device can connect with an access network in order to communicate data for the transducer with a server. The connection with the access network can be via wireless and a radio, or could be through a wired connection such as Ethernet. The device, transducer, and monitored unit can have identifiers such as bar codes, QR codes, MAC addresses, serial numbers, or other identifiers. The device can be produced by a device manufacturer and include a secure processing environment. The secure processing environment can contain a processor and nonvolatile memory, where the manufacturer records a certificate for the device with a PKI public key, cryptographic parameters associated with the certificate, and a private key. The secure processing environment can also include a set of cryptographic algorithms in order to process cryptographic operations such as a key exchange algorithm, a symmetric ciphering algorithm, and also a digital signature algorithm. The secure processing environment can also record a series of certificate authority certificates, such as a certificate authority certificate for the certificate of the device and a root certificate for the certificate authority, plus any intermediate certificates between the certificate authority certificate and the root certificate.

A device owner or device user may prefer the device transition from a manufactured state or delivered state to an operational state, where the transition for the device can comprise a configuration step. Or, after initial operation of the device, a device owner or device user may prefer the device transition from a first configured state to a second configured state, and the transition could also comprise a configuration step. In general, the security of a configuration step can be important because the overall configuration of a system can depend on the security of a configuration step. The configuration step can preferably be completed in a relatively automated manner using secure steps and hardware that is both (i) available for low cost and also (ii) readily easy to obtain or source. A mobile handset such as a smartphone could be used to connect a device

in the manufactured state to an IP network, such that the device in the manufactured state could securely communicate with a reporting system after conduction a configuration step with a configuration system.

5 The configuration step with the configuration system can comprise a first collection of steps for the mobile handset to work in conjunction with the configuration system to prepare for authentication of the device and then configuration with the device. The first collection of steps may be depicted in Figure 2a below. A mobile handset can be operated by a configuration user, which could also be the device user. The first collection of steps can comprise the mobile handset (i) reading a tag or code for the device, where the tag
10 includes a URL and an identity for the device, (ii) sending the tag to a discovery server and (iii) receiving a response from the discovery server with configuration data for provisioning the device with service, credentials, and software. The configuration data can include a URL for an authentication server and the name or location of configuration software (or “configuration app” for the mobile handset operating system).

15 The mobile handset can use the configuration software in order to establish communication between the device and a configuration system. The mobile handset can download the configuration app if not already installed on the mobile handset. The configuration data can also include a certificate for the authentication server and authentication parameters. The mobile handset can conduct a first authentication with the authentication server, where the first authentication comprises the configuration user
20 authenticating with the authentication server. The mobile handset can send device identity information to the authentication server and the authentication server can receive a certificate for the device. The authentication server can send a random number and a signature for the random number to the mobile handset.

25 The configuration step with the configuration system can also comprise a second collection of steps for the device to communicate with the mobile handset and the mobile handset to communicate with the configuration system. The second collection of steps may be depicted in Figure 3 below. The mobile handset and device can establish a NFC peer-to-peer session using the NFC radios in each, and the NFC radio for the device can be
30 dedicated to the secure processing environment for the device. Benefits from using an NFC session include the radio cost can be low and security is higher due to the close physical proximity for the device and mobile handset. The mobile handset can forward the random number and the signature for the random number to the device over the NFC session, as well as a certificate for the authentication server. The device can verify the
35 signature and the certificate. The device can sign the random number received and also generate a second random number.

Continuing with the second collection of steps in a configuration step, the device can send (i) the signature for the random number from the authentication server and (ii) the second random number to the mobile handset, which can forward to the authentication server. The authentication server can sign the second random number, and verify the signature received from the device. The device can verify the signature for the second random number from the authentication server. In this manner, both the authentication server and the device can be mutually authenticated. The authentication server can send the second random number received from the device to a configuration server. The mobile handset can collect a list of identities, which can include an RF sweep analysis to measure all available wireless networks as the location of the device in the form of a networks available list. The mobile handset can send a list of identities to the configuration system pertaining to the device, transducers, and monitored unit, and the configuration system can record the list of identities in a configuration database.

The configuration step with the configuration system can also comprise a third collection of steps for data transferred between the device and the configuration server. The third collection of steps may be depicted in Figure 4a below. A configuration server in the configuration system can also sign the second random number in order to authenticate the configuration server. The configuration server can generate an ephemeral PKI key pair and sign the ephemeral public key, and the data can be sent to the mobile handset in order to forward to the device. The device can verify the signature using a public key in a certificate for the configuration server. The configuration server and the device can use the device PKI key pair plus the ephemeral PKI key pair from the configuration server in a key exchange algorithm in order to mutually derive a symmetric encryption key.

The device can generate a new PKI key pair and send a first ciphertext to the configuration server, where the first ciphertext includes encrypted plaintext of initial configuration information for the device plus the newly derived public key. The device can encrypt the first ciphertext using the derived symmetric encryption key and the configuration server can decrypt the first ciphertext using the derived symmetric encryption key. The configuration server can have a certificate authority sign the newly derived public key for the device. The configuration server can use the identities list to collect a software package for the device. The configuration server can send the new certificate for the device, plus configuration files and network access credentials to the mobile handset in the form of a second ciphertext, and the mobile handset can forward the second ciphertext to the device. The configuration server can send a set of Wi-Fi access point credentials, in an encrypted format, to both the mobile handset and the device via the mobile handset.

The configuration step with the configuration system can also comprise a fourth collection of steps for data transferred between the device and the configuration server. The fourth collection of steps may be depicted in Figure 5 below. The device can receive the new certificate for the device as well as the configuration files from the configuration server via the mobile handset. The device can apply the configuration files and begin using the new certificate and associated derived private key in subsequent cryptographic operations. After receipt of the Wi-Fi access point credentials, the mobile handset can activate a Wi-Fi access point using the credentials, and the Wi-Fi access point may be operated at low power due to the proximity of the device. The device can activate a Wi-Fi client also using the credentials for the Wi-Fi access point received from the configuration server. The configuration server can conduct a key exchange using the new, second, derived public key for the device and the ephemeral private key in order to derive a second symmetric encryption key. The device can conduct the corresponding key exchange in order to derive the second symmetric encryption key.

The configuration server can encrypt the software package using the second symmetric encryption key and transfer to the device via the mobile handset. The device can decrypt the software package and apply the files and reboot, using both the configuration files and software package. The device can connect with the access network using the received network access credentials, and conduct a key exchange with a reporting system. The device can subsequently transfer encrypted transducer data and a report regarding configuration to the reporting system, thereby completing the configuration step for the device. The mobile handset can check with the reporting system that transducer data has been successfully transmitted between the reporting system and the device, and display to the end user that the configuration step is successful and completed.

These as well as other aspects and advantages will become apparent to those of ordinary skill in the art by reading the following detailed description, with reference where appropriate to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Various exemplary embodiments are described herein with reference to the following drawings, wherein like numerals denote like entities.

5 Figure 1a is a graphical illustration of an exemplary system, where a device with a transducer communicates with a configuration system and a reporting system, in order to conduct a configuration step, in accordance with exemplary embodiments;

 Figure 1b is a graphical illustration of hardware, firmware, and software components for a device, including a secure processing environment, in accordance with
10 exemplary embodiments;

 Figure 2a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a mobile handset, in accordance with exemplary embodiments;

 Figure 2b is a flow chart illustrating exemplary steps for creating and verifying a
15 digital signature using PKI keys, parameters, and data input, in accordance with exemplary embodiments;

 Figure 3 is a simplified message flow diagram for an exemplary system with exemplary data sent and received by a mobile handset and a device, in accordance with exemplary embodiments;

20 Figure 4a is a simplified message flow diagram for an exemplary system with exemplary data sent and received by a mobile handset and a device, in accordance with exemplary embodiments;

 Figure 4b is a series of flow charts illustrating exemplary steps for a device and a configuration server to derive keys and then use the derived keys to encrypt and decrypt
25 data, in accordance with exemplary embodiments;

 Figure 4c is a series of flow charts illustrating exemplary steps for a device and a configuration server to derive keys and then use the derived keys to encrypt and decrypt data, in accordance with exemplary embodiments;

30 Figure 4d is an illustration of a certificate for a device public key, where the device derived the public key and a corresponding private key, in accordance with exemplary embodiments;

 Figure 5 is a simplified message flow diagram for an exemplary system with exemplary data sent and received by a mobile handset and a device, in accordance with exemplary embodiments;

Figure 6a is a graphical illustration of components for a device, in accordance with exemplary embodiments, and;

Figure 6a is a graphical illustration of components for a device, in accordance with exemplary embodiments.

**DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS OF THE
INVENTION**

Figure 1a

5 Figure 1a is a graphical illustration of an exemplary system, where a device with a transducer communicates with a configuration system and a reporting system, in order to conduct a configuration step, in accordance with exemplary embodiments. The system 100 can include a device 101, a configuration system 114, and a reporting system 120. Device 101 can communicate with the configuration system 114 and reporting system 120 via
10 Internet Protocol (IP) network 128. IP network 128 could be a public or private network supporting Internet Engineering Task Force (IETF) standards such as, but not limited to, such as, RFC 786 (User Datagram Protocol), RFC 793 (Transmission Control Protocol), and related protocols including IPv6 or IPv4. A public IP network 128 could utilize globally routable IP addresses, and a private IP network 128 could utilize private IP
15 addresses which could also be referred to as an Intranet. Other possibilities for IP Network 128 exist as well without departing from the scope of the invention. Device 101 could utilize an access network 126 in order to communicate with the reporting system 120 using the IP network 128. After a configuration step 103 described in the present invention and in multiple figures below, device 101 can communicate transducer data 125 with reporting
20 system 120 through access network 126. Note that the security of transducer data 125 for device owner 122, device user 101a, and reporting system 120 during operation of device 101 may generally depend on the security of a configuration step 103.

 Access network 126 could be either a Local Area Network (LAN) or a Wide Area Network (WAN), or potentially a combination of both. Device 101 and access network
25 126 can utilize a variety of wireless technologies to communicate, including IEEE 802.11 (Wi-Fi), Low Power Wide Area (LPWA) technology, 3rd Generation Partnership Project (3GPP) technology such as, but not limited to, 3G, 4G Long-Term Evolution (LTE), or 4G LTE Advanced, NarrowBand – Internet of Things (NB-IoT), LTE Cat M, proposed 5G networks, and other examples exist as well. A wired device 101 can connect to the access
30 network 126 via a wired connection such as, but not limited to, an Ethernet, a fiber optic, or a Universal Serial Bus (USB) connection (not shown). Device 101 could be powered via any of (i) traditional “wall power” potentially with an AD/DC adapter, (ii) a battery which may be periodically recharged, (iii) power over a wired LAN connection such as “power over Ethernet”, and other possibilities exist as well.

Device 101 can include manufactured secure processing environment (SE) 102. The manufactured secure processing environment 102 can also be referred to as a secure enclave or secure element, which will also be designated herein as “SE 102”. Examples of SE 102 as of 2018 can include the secure processing environment in Figure 3 of the ARM
5 ® document “Platform Security Architecture Overview” (v. 1.1) which is incorporated by reference. Other examples for a secure processing environment exist as well, including a “Trusted Execution Environment” according to the standards promoted by Global Platform. SE 102 can comprise functionality of a processor such as an ARM or Intel ® based processor to secure cryptographic key materials including private keys in public key
10 infrastructure (PKI) key pairs, secret shared keys, cryptographic parameters, cryptographic algorithms, a certificate 107 for the secure processing environment certificate authority, a root certificate 109, etc. In other words, although root certificate 109 is depicted as external to SE 102 in Figure 1a, in exemplary embodiments a copy of this root certificate 109 is stored within nonvolatile memory of manufactured SE 102 and configured SE 104.
15 Additional details for components within a SE 102 are provided below in Figure 1b.

As depicted in Figure 1a, the configuration step 103 can convert the manufactured SE 102 into a configured SE 104. In addition, although not depicted in Figure 1a, a configuration step 103 can also be applied at a subsequent time on a previously configured SE 104. In other words, a configuration step 103 may take place multiple times over the
20 life of device 101, such as either (i) when device owner 122 changes or (ii) a certificate for device 101 (e.g. certificate cert0.SE 102k depicted below in Figure 1b) expires. An exemplary embodiment for a first configuration 103 of a device 101 is depicted in Figure 1a.

Device 101 can also include a transducer 105, and may also be referred to herein as
25 a “transducer device 101”. Transducer 105 can be a sensor or an actuator and may be either passive or active. Transducer 105 could be internal within the physical housing of device 101, such as (i) a digital image sensor inside a camera. Transducer 105 could be external to the physical housing of device 101, such as a thermocouple or probe extending from device 101 to a monitored unit 106. Additional details for transducer 105 are also
30 provided in Figure 1b below. Although device 101 is depicted in Figure 1a with a single transducer 105, device 101 could include multiple transducers 105. In addition, although a single device is depicted in Figure 1a, a system 100 can include a plurality of devices 101.

Examples of a monitored unit 106 can include an ATM or vending machine, a truck or automobile, a refrigerated or standard (“dry”) shipping container, or industrial
35 equipment such as, but not limited to, an oil field pump, a transformer connected to an electrical grid or a escalator in a building. Additional examples of a monitored unit 106

include can also include a pallet for shipping or receiving goods, a refrigerator with food, a health monitoring device attached to a person such as, but not limited to, a heart monitor, and a gate or door for opening and closing. Device 101 can utilize a sensor to measure and collect data regarding a parameter of monitored unit 106 such as, but not limited to, 5 temperature, humidity, physical location potentially including geographical coordinates from a Global Positioning System (GPS) receiver, surrounding light levels, surrounding RF signals, vibration and/or shock, voltage, current, and/or similar measurements.

Monitored unit 106 could also be controlled by device 101 via a transducer 105 that is an actuator, where device 101 can change the actuator in order to change a state for 10 monitored unit 106. For example, if monitored unit 106 is a door, then a transducer 105 could include a relay to activate a lock for the door. If monitored unit 106 is a lighting system in a building, then transducer 105 could comprise a switch to turn the lighting level up or down. Other examples exist for monitored unit 106 as well, and the above are intended to be illustrative instead of limiting. In the case where transducer 105 is either an 15 actuator or a sensor, (X) device 101 could connect transducer 105 to the reporting system 120, and (Y) reporting system 120 can both (i) receive transducer data 125 input from sensors with monitored unit 106 and (ii) transmit transducer data 125 output to actuators in order to control monitored unit 106. Thus, in operation and after the configuration step 103, reporting system 120 can remotely monitor and control monitored unit 106 using 20 device 101 and transducer 105.

As depicted in Figure 1a, device 101 can include a device user 101a, a device manufacturer 101x, and a device owner 122. Device user 101a could be a person, group of people, or business entity associated with the operation of device 101. If device 101 monitors a building, then device user 101a could be a building engineer or building 25 manager. If device 101 monitors a home, then device user 101a could be a home owner. Device manufacturer 101x can be the manufacturer or supplier of device 101 to device user 101a. Device manufacturer 101x could purchase the manufactured secure element 102 and integrate it into a housing plus other components in order to produce device 101. In exemplary embodiments, device owner 122 is the owner of device 101. Device owner 122 30 may be the same as device user 101a in some embodiments, but device owner 122 also may be different than device user 101a in other embodiments. For exemplary health care applications in a hospital where device 101 is a medical device to monitor patient health, device owner 122 could be the hospital and device user could be a nurse, doctor, or medical technician. Device owner 122 may have a certificate 123 from a certificate authority 35 associated with the device owner 122.

Each of the different entities depicted for system 100 of device owner, device user, and device manufacturer may control or operate device 101 and SE 102 at different times for the life cycle of device 101. In addition, a device owner or device user may change during the life of device 101 after manufacture and some initial configuration step that could be different than a configuration step 103. This change in control and operation over time, plus potential prior operation in an insecure manner, can create challenges for ensuring secure operation of device 101. A prior party in control of device 101 may not have supported the security procedures or requirements for device owner 122. For example, device owner 122 may require that device 101 is configured after delivery from device manufacturer 101x or possibly after delivery from a previous device owner 122. A configuration step 103 with configuration system 114 can ensure that device 101 is configured to the standards or requirements of device owner 122, without depending on previous security procedures or steps of device manufacturer 101x or device user 101a.

In order to conduct a configuration step 103, system 100 can utilize a configuration system 114. As depicted in Figure 1a, configuration system 114 can include a configuration user 108a, a mobile handset 108, a configuration application 108b, a discovery server 110, an authentication server 111, a configuration server 112, a first configuration database 112a, and a second configuration database 112b. Discovery server 110 can include a discovery server database 110a. A configuration step 124 can convert the first configuration database 112a into the second configuration database 112b, where the second configuration database 112b can record data pertaining to a device 101 and reporting system 120 after a configuration step 103. The configuration step 124 can comprise the server/network-side equivalent step of configuration step 103 depicted for device 103. The elements within a configuration system 114 can be connected via a configuration network 113. Configuration network 113 could be similar to IP network 128 described above.

Configuration system 114 can utilize a certificate 115 for a certificate authority associated with configuration system 114. The certificate authority associated with configuration system 114 may or may not be the same as the certificate authority associated with SE 102, or owner 122, or reporting system 120. Although a certificate authority is not depicted for certificate 115 in Figure 1a, each of the certificates 107, 115, 121, and 123 preferably have a "parent" certificate authority or signed public key associated with the certificates. Certificates 107, 115, 121, and 123 can be formatted similar to SE certificate 419 depicted in Figure 4d below, with different values appropriate for each certificate depicted in Figure 1a. Root certificate 109 may also be similar to SE certificate 419,

except the signature in a root certificate 109 may be self-signed by the public key in root certificated 109.

5 A configuration system 114 can use a mobile handset 108 operated by a configuration user 108a. Details and components for a mobile handset 108 are depicted and described below in connection with Figure 2a. In some exemplary embodiments, a mobile handset 108 can comprise a smart phone, such as based on the Android or IOS operating systems. In other embodiments, instead of using a “mobile handset 108”, a configuration user 108a can operate a “configuration unit 108”, or simply “unit 108”, which can provide the same or equivalent functionality as a “mobile handset 108” as depicted herein. In these other exemplary embodiments, a unit 108 could comprise a portable device, such as laptop, tablet, wearable device such as a smart watch, a USB stick, etc. For embodiments where device 101 is configured in a manufacturing facility or potentially a location away from monitored unit 106, including the location where SE 102 is inserted into device 101, then the element or node depicted as “mobile handset 108” in the present invention could comprise the unit 108 where the unit 108 could operate normally in a fixed location instead of being mobile. For example, unit 108 could be a configuration server to conduct a configuration step 103 for a plurality of devices 101 in sequence. Other possibilities exist as well for a unit 108 that is not a mobile handset to provide the functionality of a “mobile handset 108” herein and below, without departing from the scope of the present invention.

The servers shown for configuration system 114 can be co-located within the same data center or geographically dispersed. A configuration system 114 can also include a plurality of the elements depicted in Figure 1b. Further, some elements of a configuration system 114 could be combined, such (i) as the discovery server 110 could be combined with the authentication server 111 or configuration server 112, or (ii) the configuration server 112 could be combined with the authentication server 111. Other possibilities exist for the arrangement of elements within a configuration system 114 without departing from the scope of the invention. For an alternative exemplary embodiment to that depicted in Figure 1a, discovery server 110 could be operated external to configuration system 114, such as discovery server being associated with device manufacturer 101x, device owner 122, or reporting system 120. Additional details regarding the components and operation of a configuration system 114 will be described below. In some exemplary embodiments, reporting system 120 and configuration system 114 can be combined.

35 A reporting system 120 in system 100 can include a reporting server 116, an application server 117, and a reporting database 118. Although not depicted in Figure 1a, a reporting system 120 and a configuration system 114 can include a firewall to filter packets

received through an IP network 128. The elements within a reporting system 120 can be connected via reporting network 119, which could comprise a network similar to IP network 128. Similar to configuration system 114, the servers for reporting system 120 can be co-located within the same data center or geographically dispersed. The servers and
5 databases shown for both reporting system 120 and configuration system 114 can be either different physical computers such as rack-mounted servers, or different logical or virtual servers or instances operating in a “cloud” configuration.

The elements within reporting system 120 can operate in conjunction to collect data from a plurality of devices 101 through access network 126 and present data or reports to
10 device owner 122, device user 101a, or potentially another user or manager of system 100. Reporting system 120 could also take input from device owner 122 or device user 101a to send control transducer data 125 to device 101 in order to operate a transducer 105 in order to control monitored unit 106. Reporting system 120 could also send control transducer data 125 to device 101 without requiring user input, such as for automated control.
15 Reporting system 120, or elements in reporting system 120, can have a certificate 121 signed by a certificate authority associated with reporting system 120. Certificate 121 may be associated or verified using root certificate 107, although in some exemplary embodiments the root certificate for certificate 121 may be different than root certificate 107. The various servers depicted in Figure 1a could comprise multiple individual physical
20 servers or multiple virtual servers operating in a coordinated manner to provide the functionality shown. In other words, although a single reporting server 116 is depicted in Figure 1a, a reporting server 116 could comprise multiple different physical or virtual servers.

25 **Figure 1b**

Figure 1b is a graphical illustration of hardware, firmware, and software components for a device, including a secure processing environment, in accordance with exemplary embodiments. Figure 1b is illustrated to include several components that can be common within a device 101 and a manufactured secure processing environment 102 or
30 secure enclave 102 (SE 102). Device 101 may consist of multiple components in order to collect and transmit transducer data 125 (shown in Figure 1a) associated with a monitored unit 106. In exemplary embodiments and as depicted in Figure 1b, device 101 can include a device identity 101b, a processor 101c (depicted as “CPU 101c”), random access memory (RAM) 101d, an operating system (OS) 101e, storage memory 101f, a system bus 101g, a

transducer interface 101h, a transducer bus 101i, a radio 101z, and a user interface (UI) 101y.

Device identity 101b could comprise a preferably unique alpha-numeric or hexadecimal identifier for device 101, such as an Ethernet MAC address, an International Mobile Equipment Identity (IMEI), an owner interface identifier in an IPv6 network, a serial number, or other sequence of digits to uniquely identify each of the many different possible units for device 101. Device identity 101b can preferably be recorded in a non-volatile memory or written directly to hardware in device 101 by device manufacturer 101x upon device manufacturing. The CPU 101c can comprise a general purpose processor appropriate for typically low power consumption requirements for a device 101, and may also function as a microcontroller. CPU 101c can comprise a processor for device 101 such as an ARM® based process or an Intel® based processor such as belonging to the Atom or MIPS family of processors, and other possibilities exist as well. CPU 101c can utilize bus 101g to fetch instructions from RAM 101d and operate on the instruction. CPU 101c can include components such as registers, accumulators, and logic elements to add, subtract, multiply, and divide numerical values and record the results in RAM 101d or storage memory 101f, and also write the values to an external interface such as radio 101z.

RAM 101d may comprise a random access memory for device 101. RAM 101d can be a volatile memory providing rapid read/write memory access to CPU 101c. RAM 101d could be located on a separate integrated circuit in device 101 or located within CPU 101c. The RAM 101d can include data recorded in device 101 for the operation when collecting and communicating transducer data 125 regarding monitored unit 106. The system bus 101g may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures including a data bus. System bus 101g connects components within device 101 as illustrated in Figure 1a, such as transferring electrical signals between the components illustrated. System bus 101g can also connect SE 102 to the other elements in device 101 including CPU 101b, storage memory 101f, and transducer interface 101h. Device 101 can include multiple different versions of bus 101g to connect different components, including a first system bus 101g between CPU 101b and RAM 101d (which could be a memory bus), and a second system bus 101b between CPU 101b and transducer interface 101h, which could be an I2C bus, an SPI bus, a USB, Dallas 1 wire, or similar data busses, which could also comprise a transducer bus 101i.

Memory 101f within device 101 can comprise a non-volatile memory for storage of data when device 101 is powered off. Memory 101f may be a NAND flash memory and record firmware for device 101. Memory 101f can record long-term and non-volatile

storage of data or files for device 101. In an exemplary embodiment, OS 101e is recorded in memory 101f when device 101 is powered off, and portions of memory 101f are moved by CPU 101b into RAM 101d when device 101 powers on. Memory 101f (i) can be integrated with CPU 101b into a single integrated circuit (potentially as a “system on a chip” as depicted in Figure 6b below), or (ii) operate as a separate integrated circuit or a removable card or device, such as a removable SD card. Memory 101f may also be referred to as “device storage” and can include exemplary file systems of FAT16, FAT 32, NTFS, ext3, ext4, UDF, or similar file systems. As depicted in Figure 1b, non-volatile memory 101f may also contain SE Memory 102x. SE Memory 102x can be a portion of memory 101f that is allocated to SE 102. In exemplary embodiments, the two sets of memory 101f and 102x can be segmented logically, where memory 102x is encrypted with a symmetric ciphering key 102z within SE 102 by a memory controller for SE 102 (not shown). SE 102 could use a symmetric ciphering algorithm such as the Advanced Encryption Standard (AES-128), AES-192, Triple Data Encryption Algorithm (3DES), or similar algorithms. In other words, CPU 101c would not normally be able to read plaintext data in memory 102y since that memory is encrypted by SE 102.

The operating system (OS) 101e can include Internet protocol stacks such as a User Datagram Protocol (UDP) stack, Transmission Control Protocol (TCP) stack, a domain name system (DNS) stack, etc., and the operating system 101e may include timers and schedulers for managing the access of software to hardware resources within device 101, including SE 102 and transducer interface 101h. The operating system shown of 101e can be appropriate for a low-power device with more limited memory and CPU resources (compared to a server such as reporting server 116 and configuration server 112). Example operating systems 101e for a device 101 includes Linux, Android® from Google®, iOS from Apple®, Windows® 10 IoT Core, or Open AT® from Sierra Wireless®. Additional example operating systems 101e for a transducer device 101 include eCos, uC/OS, LiteOs, Contiki, OpenWRT, Raspbian, and other possibilities exist as well without departing from the scope of the present invention. Although depicted as a separate element within device 101 in Figure 1b, OS 101e may reside in RAM 101d or memory 101f during operation of device 101.

Device 101 can include a radio 101z to communicate wirelessly with networks such as access network 126 depicted and described in Figure 1a above. Radio 101z could connect with an antenna in order to transmit and receive radio frequency signals. Although not depicted in Figure 1b, device 101 could utilize a wired connection such as Ethernet for external communication instead of or in addition to a radio 101z. Device 101 may also optionally include user interface 101y which may include one or more devices for receiving

inputs and/or one or more devices for conveying outputs. User interfaces are known in the art and may be simple for many devices such as a few LED lights or and LCD display, and thus user interfaces are not described in detail here. User interface 101y could comprise a touch screen if device 101 as more sophisticated interaction with user 101a. Device 101
5 can optionally omit a user interface 101y, if no user input or display is required for operating device 101 with monitored unit 106. Although not depicted in Figure 1b, device 101 can include other components to support operation, such as a clock, power source or connection, antennas, etc.

Device 101 can include a manufactured secure processing environment (SE) 102.

10 SE 102 may include separate components of a processor SE CPU 102a, RAM 102b, a Near Field Communications (NFC) radio 102c, an SE external bus controller 102h, a hardware random number generator 102e, an SE flash memory 102d, an SE internal bus 102i, a transducer input/output bus controller 102w, and an electrically erasable programmable read-only memory (EEPROM) 102j. In exemplary embodiments, the NFC radio 102c may
15 support other wireless standards besides NFC (such as Wi-Fi or Bluetooth), and in those embodiments an “NFC radio 102c” can be referenced herein as a “radio 101z”. Components for SE 102 such as CPU 102a, RAM 102b, radio 102c, flash memory 102d could be similar to components described above for device 101 such as CPU 101c, RAM 101d, radio 101z, and storage 101f, respectively, except the components for SE 102 could
20 have lower capacity, bandwidth, functionalities, or costs compared to the equivalent components for device 101. Although SE 102 is depicted as a separate component within device 101, in some embodiments the functionality of SE 102 described below could be implemented in software or firmware, such that a physical SE 102 can be omitted and a device 101 can perform the functions and operations of an SE 102 as described herein.

25 In a first exemplary embodiment, (i) CPU 101c may operate at greater than 200 MHz, while SE CPU 102a may operate at less than 200 MHz, and (ii) RAM 101d may store more than two megabytes while RAM 102b may store a value equal to or less than two megabytes. SE CPU 102a can comprise a processor similar to CPU 101c, but embedded into SE 102 and consequently typically with less resources than a CPU 101c,
30 such as (i) less cache memory, (ii) operating typically at a slower speed, (iii) fewer internal registers, (iv) lower power consumption compared to CPU 101c, etc. In some exemplary embodiments, SE CPU 102a can comprise a processing core within a multi-core processor CPU 101c, and in those embodiments SE CPU 102a could operate with the same or similar clock rates and memory as other cores within CPU 101c.

35 In a second exemplary embodiment, although depicted as separate elements for both (i) CPU 101c, and RAM 101d, and (ii) SE CPU 102a and SE RAM 102b, the two

elements could comprise the same physical hardware but represent time division or time separation of the physical hardware, similar to how the same physical hardware can host multiple logical processing running concurrently. In other words, CPU 101c could switch operations between a “normal” mode functioning as CPU 101c for device 101 and then a “secure” mode as SE 102 for device 101. Or, SE CPU 102a could represent a virtualized process or computer within device 101, where the OS 101e implements virtualization and operates as a host for SE 102. Switching could take place rapidly enough that CPU 101c and SE CPU 102a would appear to other components both inside and outside device 101 as separate processors, while there could be a single physical processor inside device 101.

As depicted in Figure 1b, SE 102 can include a radio 102c, which could support NFC communication. Although illustrated as internal to SE 102, radio 102c could be external to SE 102 in some embodiments. A radio 102c that is external to SE 102 could be connected via a secure bus such as SE bus 102i. In some embodiments, SE bus 102i can extend outside SE 102 in order to connect external components for SE 102 with internal components for SE 102. Radio 102c can comprise a radio for short-distance wireless communication, and be connected with an antenna. Radio 102c, when operating to support NFC communications, could (i) transmit and receive at the globally available frequency for unlicensed use of 13.56 MHz, and (ii) be compatible with the International Standards Organization (ISO) 14443 standards and subsequent and related versions. Radio 102c can operate as any of an NFC reader, NFC writer, and NFC peer-to-peer communication, and preferably supports communication at short range such as within several inches of device 101. Radio 102c can support radio frequency communications with mobile handset 108, when mobile handset 108 is within close physical proximity of device 101, such as less than a few feet in an exemplary embodiment. An exemplary processor with an embedded NFC radio as of early 2018 would be the PN7360AUHN processor from NXP ® Semiconductor. Alternatively, a regular processor (i.e. without a radio 102c) could be combined with a radio controller on a separate chip such as combining an ARM Cortex ® processor with a radio chip like the CLRC663 from NXP ® Semiconductor. Other possibilities for the configuration of a radio 102c with a SE 102 and device 101 are possible in order to support a configuration step 103 without departing from the scope of the present invention.

A benefit of the use of short-range communication for SE 102 is physical security, such that any external device desiring to communicate with SE 102 through radio 102c must be in close physical proximity. In exemplary embodiments radio 102c could support other short-range wireless standards besides NFC. Radio 102c could implement radio frequency protocols that utilize low power and close proximity for the other node for which

SE 102 will communicate with. In exemplary embodiments, radio 102c could be a Bluetooth or Wi-Fi radio, but operate at intentionally reduced in order to require closer physical proximity an external device such as mobile handset 108 to communicate with SE 102. For embodiments where radio 102c operates as a Bluetooth or Wi-Fi radio, radio
5 102c could transmit at a power in an exemplary range of 0.01 – 1.0 watts. Other possibilities for the radio technology and power levels of radio 102c exist without departing from the scope of the present invention.

SE 102 can also include a SE external bus controller 102h, which can manage the sequence and flow of data between SE 102 and device 101. As depicted in Figure 1b, SE
10 external bus controller 102h could electrically connect the SE internal bus 102i with device bus 101g, such that electrical signals, pulses, or waveforms can be transferred between device 101 and SE 102, including the transfer of binary data. SE external bus controller 102h could manage and sequence the flow of data between the two sides, and include a data buffer and logic gates. SE external bus controller 102h can include security
15 components, such that device 101 cannot feasibly read data from EEPROM 102j or SE RAM 102b. SE external bus controller 102h could operate as a “mailbox”, such that (i) device 101 writes values or data to SE external bus controller 102h and then (ii) SE 102 reads the values or data from SE external bus controller 102h.

As illustrated in Figure 1b, device 101 may also contain a random number
20 generator 102e. Random number generator 102e can comprise a physical component that generates random or pseudo random numbers for cryptographic operations of SE 102. A random number generator 102e can also comprise a hardware random number generator. The creation of random numbers with a high degree of entropy is important in the use of subsequent steps such as generating Random1.SE 308 below and also deriving PKI keys
25 such as step 401 below. A seed for random number generator 102e could comprise a plurality of data from within SE 102 appended together in order to accumulate information entropy. To acquire the seed, SE 102 could collect a plurality of transducer 105 measurements or states, radio 102c measurements, clock times or values for CPU 102a, RAM 102b or memory 102x states, etc. In exemplary embodiments, random number
30 generator 102e can include a secure hash algorithm similar to message digest 230 below in Figure 2b operating on the random number seed. In exemplary embodiments, random number generator 102e operates within SE 102 as depicted in Figure 1b, and in this manner the random numbers used for cryptographic operations such as PKI key generation in a step 401 below in Figure 4a can remain reasonably secure and not normally communicated
35 outside SE 102.

SE 102 can include nonvolatile SE flash memory 102d, which can also be referred to as “nonvolatile memory 102d” or “memory 102d”. Nonvolatile memory 102d can comprise a physical memory of NAND or NOR flash memory, such that data recorded in nonvolatile memory 102d continues to be recorded when electrical power is removed from device 101 or SE 102. The data within non-volatile memory 102d can subsequently be read and/or re-written when power is restored to SE 102. Memory 102d for SE 102 can be similar to memory 101f for device 101. Memory 102d can be configured for a file system, such as those listed for memory 101f above.

As depicted in Figure 1b, memory 102d can include config0.SE 102aa. Config0.SE 102aa can comprise a file or set of files with the default configuration of SE 102, such as enabling or disabling radio 102c upon startup, settings for radio 102c and credentials, and parameters for the other elements of SE 102 depicted in Figure 1b. Config0.SE 102aa can also be referred to as an initial device configuration and include values for a device configuration. In an exemplary embodiment, config0.SE 102 comprises the firmware and software installed on device 101 upon manufacturing, similar to software package 104q below in Figure 5, where software package 104q can provide updated software, firmware, and configuration files for device 101 and config0.SE contains the originally installed versions of these files in device 101. In an exemplary embodiment where radio 102c operates as a Wi-Fi radio, config0.SE 102aa can include values such as an SSID, a password or security key, and a user name, or also multiple sets of those and other values.

SE 102 can also include a transducer input/output bus controller 102w (or “TR controller 102w”), which can manage the sequence and flow of data or signals between (a) SE 102 and transducers 105 via (b) transducer bus 101i and transducer interface 101h. As depicted in Figure 1b, TR controller 102w could electrically connect the SE internal bus 102i with transducer bus 101i, such that electrical signals, pulses, or voltages can be transferred between transducers 105 and SE 102, including the transfer of digital data or analog signals. For digital data, TR controller 102w could manage and sequence the flow of data between the two sides, and include a data buffer and logic gates. For analog data, TR controller 102w could include a digital-to-analog converter (DAC) or an analog-to-digital converter (DAC). TR controller 102w can include security measures, such that transducers 105 cannot feasibly read data from EEPROM 102j or SE RAM 102b.

In an alternative exemplary embodiment to that depicted in Figure 1b, SE 102 can use a second, different bus 102i than the depicted bus 102i in order to connect with TR controller 102w. In this alternative exemplary embodiment, external transducers 105 could be physically and electrically separated from secure information such as cryptographic keys or data in memory or the processor. Although TR controller 102w is depicted in Figure 1b

as internal to SE 102 and connected to SE internal bus 102i, in another exemplary embodiment TR controller 102w could be connected to device bus 101g, and SE 102 could communicate with TR controller 102w via SE external bus controller 102h. Other possibilities exist as well for the configuration and connections of TR controller 102w without departing from the scope of the present invention.

Although not depicted in Figure 1b, the various servers shown above in Figure 1a such as configuration server 112 and reporting server 116 and other servers as well can include equivalent internal components as device 101 in order to operate. The servers in Figure 1a could include a processor similar to CPU 101c, with primary differences for the processor server being increased speed, increased memory cache, an increased number and size of registers, the use of a 64 bits for datapath widths, integer sizes, and memory address widths, as opposed to an exemplary 32 or 16 bits for CPU 101c in device 101. Similarly, RAM in a server could be a RAM similar to RAM 101d in device 101, except the RAM in a server could have more memory cells such as supporting exemplary values greater than an exemplary 2 gigabytes, while RAM in device 101 could support fewer memory cells such as less than an exemplary 1 gigabyte. Non-volatile memory for storage in a server in Figure 1a could comprise disks, "solid state drives" (SSDs) or "storage area networks" (SAN) for a server. Instead of a radio 101z, in exemplary embodiments, a server in Figure 1a could use a physical, wired LAN interface such as a high-speed Ethernet or fiber optic connection.

As depicted in Figure 1b, a device 101 or SE 102 can communicate with transducers 105 via a transducer interface 101h. Transducer interface 101h can comprise the physical interface between transducers 105 and device 101, and include electrical connectors or pins for transfer of electrical signals between device 101 and transducers 105. For embodiments where a transducer 105 utilizes a micro universal serial bus (USB), then transducer interface 101h could comprise a micro USB receptacle. Transducer interface 101h can utilize a transducer bus 101i. For digital data acquisition or control of transducers 105, transducer bus 101i could comprise a data bus similar to bus 101g for device 101 or bus 102i for SE 102 as described above. Exemplary buses for digital data include universal serial bus (USB), serial peripheral interface (SPI), inter-integrated circuit (I2C), and Dallas 1-wire. Transducers 105 could also be connected via wireless protocols such as Bluetooth or Wi-Fi, and in this case transducer interface 101h could include a radio similar to radio 101z.

A transducer bus 101i can connect components in order to communicate with transducers 105. Exemplary components for transducer bus 101i include general purpose input/output (GPIO) controller 101k, an analog to digital controller (ADC) 101m, a digital-

to-analog controller (DAC) 101n, and another controller 101j. GPIO 101k could comprise electrical pins that are unused by default but could be programmed via instructions residing in SE RAM 102b or device RAM 101d, where the instructions set the pins on GPIO high or low depending on the transducer 105 connected or configuration of device 101. As one
5 example software or firmware operating on device 101 could program GPIO 101k to operate as a traditional serial or RS-232 interface, and other possibilities exist as well. In an exemplary embodiment, the software package 104q depicted and described below in connection with Figure 5 can include software to program GPIO 101k. ADC 101m could take analog input from transducer 105 and convert it to digital signals for further
10 processing by device 101 or SE 102. As one example, ADC 101m could read a level of a voltage input by measuring the voltage level and outputting a digital value of the measured voltage level, such as an exemplary reading of 0.123 volts.

DAC 101n could take digital input from device 101 or SE 102 and convert it to analog signals for transducer 105. As one example, DAC 101n receive a digital signal
15 from device 101 to output a voltage level, and convert the digital signal to the analog level output, such as an exemplary value of 0.123 volts. Another controller 101j could comprise a different controller than GPIO 101k, ADC 101m, and DAC 101n listed above, such as a controller customized for a particular transducer 105 or application of device 101. Another controller 101j could be a USB controller, military standard 1553 controller, peripheral
20 component interconnect express (PCIe) bus, or an Ethernet networking interface including power over Ethernet. Other possibilities exist as well for another controller 101j operating with a transducer interface 101h without departing from the scope of the present invention.

In exemplary embodiments, transducer interface 101h can connect with several types of transducers, such as (i) a transducers input 105b that could comprise a sensor, or
25 (ii) a transducer output 105c that could comprise an actuator, or (iii) a bi-directional transducer 105a. A bi-directional transducer 105a can transfer data (a) from device 101 to transducer 105a and (b) from transducer 105a to device 101. Exemplary bi-directional transducers 105a could include a radio 105aa or a payment terminal 105ab, and other examples exist as well. Radio 101aa could comprise a radio similar to radio 101z and be
30 connected with an antenna. The transducers 105 shown in Figure 1b are depicted to be illustrative as opposed to limiting. For example, another bi-directional transducer 105a could be a touch screen, where (x) data transmitted to the touch screen from device 101 can provide visual display information such as pixel status or color, and (y) data received from the touch screen going to device 101 is touch information such as location of manual touch
35 on the screen.

Transducer input 105b could support transducers that provide digital or analog data input into transducer bus 101i. Exemplary devices for transducer input 105b depicted in Figure 1a include a probe 105ba, a keypad 105bb, a sensor 105bc, and a switch 105bd. In other words, transducer input 105b could receive data about monitored unit 106 and forward the data to device 101 through transducer interface 101h and transducer bus 101i. Probe 105ba could measure values for monitored unit such as pH, salinity, humidity, voltage, 3-axis orientation, acceleration, or similar physical characteristics of monitored unit 106. Keypad 105bb could be a pad for a device user 101a to enter numbers or letters, including a keyboard or a keypad for entering a PIN code on a payment terminal, or a password. Sensor 105bc could be operated in an active or passive manner. Sensor 105bc can receive an electrical current input from device 101 in an active mode. An exemplary sensor 105 could comprise a thermistor or thermocouple. A sensor 105bc could also comprise a camera for generating a picture or video from the outside of device 101. Switch 105bd could comprise a switch operated by a user 101a or an external automated process associated with monitored unit 106. For example, switch 105bd could forward data to device 101 when electrical contacts associated with switch 105bd are open or closed. Other possibilities exist as well for transducer input 105b without departing from the scope of the present invention.

Transducer output 105c could support transducers that receive digital or analog data output from transducer bus 101i. Exemplary devices for transducer output 105 depicted in Figure 1b include voltage 105ca, a display 105cb, a relay 105cc, and an actuator 105cd. Voltage 105ca could be a voltage output by device 101 for controlling monitored unit 106. For example, monitored unit 106 could include a physical interface similar to transducer interface 101h, which could receive an output voltage 105ca (e.g. output from device 101 but input into monitored unit 106). Voltage 105ca could control a level of a lighting system, a speed of a motor, or other values associated with monitored unit 106. Display 105cb could be a light emitting diode (LED), liquid crystal display (LCD), a computer screen or monitor, or similar systems of displaying information to users 101a. Relay 105cc could be a relay with electrical contacts that are opened or closed via input from device 101, such that the state of monitored unit 106 changes when relay 105cc is opened or closed. An exemplary relay 105cc could be a lock on a door. Actuator 105cd could comprise devices that receive analog or digital input from device 101 in order to change or control the state of monitored unit 106. Exemplary actuators 105cd include electric motors, hydraulic cylinders, piezoelectric actuator, a speaker, solenoids, etc.

Although transducers 105 in Figure 1b are depicted as external to device 101, transducers 105 could be internal to device 101, or device 101 could utilize a mix of

internal and external transducers 105 connected with a transducer interface 101h. Further, a device 101 could implement multiple different transducer interfaces 101h, such as a first transducer interface 101h connects internal transducers 105 and a second transducer interface 101h connects external transducers. Other possibilities exist as well without departing from the scope of the present invention.

As depicted in Figure 1b, SE 102 could also include an EEPROM 102j. EEPROM 102j could comprise long-term, nonvolatile memory storage chipsets or physical units that are designed primarily for writing once or a few times, and reading many times. As contemplated within the present invention, a read-only address within EEPROM 102j could comprise a memory address or another hardware address for read-only operations accessible via bus 102i. Although depicted as using EEPROM 102j, the data and elements depicted within EEPROM 102j in Figure 1b could be alternatively stored in flash memory 102d, where the memory within 102d is tagged or set as being “read only” during most of the life of operation of SE 102. In other words, a physical EEPROM chip or memory cells are not required for some embodiments in order to implement the functionality of EEPROM 102j depicted in Figure 1b.

EEPROM 102j could include a certificate cert0.SE 102k, firmware 102u, and secret key SK0.SE 102y, symmetric encryption key 102z and message authentication code (MAC) key 102za. Cert0.SE 102k for SE 102 can be the default, installed certificate within SE 102 and can take a form similar to SE certificate 419 depicted and described in connection with Figure 4b and Figure 4d below. A device manufacturer 101x could deliver device 101 to user 101a with certificate cert0.SE 102k pre-installed as well as SK0.SE 102y recorded in a nonvolatile memory. Device manufacturer 101x could receive SE 102 from a processor manufacturer with certificate cert0.SE 102k configured within non-volatile memory of SE 102, or device manufacturer 101x could write certificate cert0.SE 102k and corresponding SK0.SE 102y (and/or cert0.SE 102k' and SK0.SE 102y') to SE 102 during a pre-configuration step conducted before the depicted configuration step 103 from Figure 1a. SK0.SE 102y can also comprise two different values or numbers, where a first number is associated with a key exchange such as key exchange step 411a below in Figure 4a and Figure 4b, and a second value or number associated with a digital signature algorithm such as generating signatures 309 and 416 in Figure 2b. Or, the two values or numbers could be represented by separate keys SK0.SE 102y and SK0.SE 102y', respectively.

In exemplary embodiments, SE 102 can also include, record, or store a second secret key SK0.SE 102y'. The first SK0.SE 102y from the above paragraph could be used with key exchange algorithms such as ECDH key exchange step 411a in Figure 4a and

Figure 4b below. The second key SK0.SE 102y' could be used with digital signature algorithms such as those depicted for SE 102 in Figure 2b below. Likewise, SE 102 could also include, record, or store a second certificate cert0.SE 102k', where the second certificate cert0.SE 102k' could be used with digital signature algorithms and include a public key corresponding to the second secret key PK0.SE 102y'. In some embodiments, the second secret key SK0.SE 102y' and the second certificate cert0.SE 102k' could be omitted, and the key SK0.SE 102y could be used with both key exchange step 411a and digital signatures.

In addition, symmetric encryption key 102z and MAC key 102za along with firmware 102u could be written by a manufacturer into non-volatile memory of SE 102 (such as EEPROM 102j) before distribution of device 101 to users 101a. Symmetric encryption key 102z can comprise a key to encrypt and decrypt ciphertext similar to symmetric encryption key 442a depicted in Figure 4b below. Key 102z can be used with a symmetric encryption algorithm 444a depicted and described in connection with Figure 4b below. The ciphertext for key 102z can comprise firmware 102u or components of firmware 102u, such that the firmware 102u or components of firmware 102u could be securely stored outside SE 102 in an encrypted format and loaded into SE 102 after decryption using key 102z. MAC key 102za can be associated with key 102z in order to verify integrity of ciphertext encrypted or decrypted with key 102z.

Certificate cert0.SE 102k can include an identity ID.SE 102m, public key PK0.SE 102n, certificate parameters 102p, and a signature0.CA0 102q. Identity ID.SE 102m may comprise a unique identity for SE 102, in order to identify SE 102 among a plurality of devices 101 that connect with either access network 126, configuration system 114, or reporting system 120. In exemplary embodiments, ID.SE 102m remains the same for a given physical SE 102, even as new or subsequent certificates are issued for SE 102. PK0.SE 102n can comprise a public key for SE 102, and an exemplary public key within an X.509 certificate is depicted in Figure 4d below as PK1.SE 410a. In an exemplary embodiment, the ID.SE 102m is included as the field "serialNumber" (SN) in the subject of an X.509 certificate, or alternatively the ID.SE 102m could be in the "commonName" field (CN).

Certificate parameters 102p can comprise parameters associated with cert0.SE and exemplary parameters for certificate parameters 102p include the parameters 407 depicted in Figure 4b below and also the parameters 407 in Figure 4d below. Certificate parameters 102p can specify algorithms such as RSA or elliptic curve cryptography (ECC) values, ECC curve names, key lengths in bits, signature algorithms, validity dates, encoding rules, etc. Signature signature0.CA0 102q may comprise the signature of the certificate authority

for SE 102, and the public key of the certificate authority for SE 102 could be included in cert.CA0 107. A node of system 100 in Figure 1a can use cert0.SE and cert0.CA 107 to verify that the PK0.SE 102n belongs to SE 102 with identity ID.SE 102m. Additional details regarding the use of public keys, private keys, signatures, and certificates for the components in Figure 1a and Figure 1b will be discussed in figures below, including Figure 2b.

In exemplary embodiments, firmware 102u for SE 102 can include SE bootloader firmware 102t, and SE boot configuration 102v. SE bootloader firmware 102t in EEPROM 102j can provide machine executable code for SE CPU 102a to initiate operations when electrical power is provided to device 101 and SE 102. SE bootloader firmware 102t in EEPROM 102j could also include instructions for SE CPU 102a to load config0.SE recorded in nonvolatile memory 102d, if present, where config0.SE is described with SE nonvolatile memory 101d. SE bootloader firmware 102t could load boot firmware for SE 102, where the boot firmware for SE 102 could be stored in either memory 102d or memory 102x. If boot firmware for SE 102 is stored in memory 102x (where boot firmware for SE 102 is called by bootloader firmware 102t), then in exemplary embodiments boot firmware SE 102 is encrypted with key 102z since the boot firmware would be stored outside SE 102. In exemplary embodiments, bootloader firmware 102t includes a set of cryptographic algorithms 141, where cryptographic algorithms 141 as a minimum supports the use of key 102z and a symmetric decryption algorithm 444b (see Figure 4b) in order to decrypt boot firmware for SE 102 stored outside SE 102, as well as use MAC key 102za to verify integrity of a boot firmware stored in memory 102x.

Bus control libraries 142 could include software or firmware to manage and schedule the input and output of data for SE 102, such as machine code for (i) instructing processor 102a to write data to bus 102i for bus controller 102h or bus controller 102w when data is transmitted outside SE 102, (ii) read data from bus controller 102h or bus controller 102w when data from device 101 or transducer 105 is passed into SE 102, and (iii) reading SK0.SE 102y and/or SK0.SE 102y' from protected memory. Bus control libraries 142 can also include bus read instructions 142a and bus write instructions 142b. Bus read instructions 142a can provide machine executable code for SE CPU 102a to read data from either (i) device 101 using bus controller 102h, or (ii) transducer 105 using bus controller 102w. In this manner, bus read instructions 142a could provide the logical software or firmware interface for SE 102 to receive external data. Bus read instructions 142a could specify memory addresses, registers, or bus addresses where device 101 or transducers 105 can write data in order to be read by SE 102. Bus write instructions 142b

can provide machine executable code for processor 102a to write data output from SE 102 in order to be read by device 101 or transducer 105.

Firmware 102u in SE 102 for device 101 can also include a SE boot configuration 102v, a certificate authority public key recorded in cert.CA0 107 and a root certificate cert.CA.root 109, as depicted in Figure 1b. SE boot configuration 102v can provide values for the configuration or operation of SE 102 when operating with the SE bootload firmware 102t, such as (i) parameters for using bus 102i or controllers 102h and 102w, (ii) the frequency to operate a clock or phase looped logic associated with SE 102, (iii) a firmware version number, (iv) the memory capacity of memory 102d and 102x, (v) the memory addresses, cells, or file sectors to utilize for bus controllers, and (vi) parameters specifying values for hardware within SE 102. In exemplary embodiments, SE boot configuration 102v specifies the activation of radio 102c, such that if (i) a configuration step 103 has not been conducted to configure SE 102, then radio 102c is active, but (ii) if a configuration step 103 has been conducted then SE boot configuration 102v specifies that radio 102c is inactive or turned off. Note that config.SE 102aa could specify the activation of radio 102c instead of SE boot configuration 102v. SE boot configuration 102v may specify the parameters for use of boot firmware 102u, while parameters config0.SE 102aa may specify parameters for operating firmware of SE 102 that is loaded by boot firmware 102t. Cert.CA0 107 and cert.CA.root 109 were also described above in connection with Figure 1a. Note that by recording cert.CA0 107 and cert.CA.root 109 in EEPROM 102j for SE 102, including during manufacturing, these certificates for the certificate authority can be securely distributed along with SE 102 for subsequent use such as verifying signatures as described below.

Figure 2a

Figure 2a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a mobile handset, in accordance with exemplary embodiments. System 200 can include a mobile handset 108, device 101, a discovery server 110 and authentication server 111. Mobile handset 108 can communicate with discovery server 110 and authentication server 111 via IP network 128. Mobile handset 108 can comprise a smart phone such as a phone based on an Android operating system from Google ® or IOS from Apple ®. Mobile handset 108 can include a battery, radio, and touch screen, as well as a camera 108r. Mobile handset 108 can connect with an access network similar to access network 126 in order to communicate with discovery server 110 and authentication server 111, and the access network used in a system 200 can be a different access network than access network 126 used by device 101 in Figure 1a and

Figure 5. Device 101 in system 200 can comprise a device 101 depicted in Figure 1a and Figure 1b, and device 101 can also include a tag 101s. Tag 101s can comprise a recorded image on device 101 or packaging of device 101 and could be a bar code, QR code, or a serial number for device 101. Tag 101s can provide a unique identifier for device 101. In
5 exemplary embodiments, if device 101 is a medical device, then tag 101s can correspond to a Unique Device Identification (UDI).

Although Figure 2a depicts a mobile handset 108, a configuration unit as described above in Figure 1a could be utilized in system 200 instead of a mobile handset 108. The configuration unit could perform equivalent steps and conduct equivalent message flows as
10 depicted in Figure 2a, Figure 3, and related figures below. In other words, a configuration unit could be a device that is different than a regular mobile handset 108, but perform similar steps and provide similar functionality. In an exemplary embodiment, mobile handset 108 could be a configuration unit customized for operation in systems 100, 200, 300, etc. Mobile handset 108 is depicted for preferred exemplary embodiments because
15 mobile handset 108 could be a relatively ubiquitous piece of equipment for a device user 101a or configuration user 108a, but other equipment could provide equivalent functionality as mobile handset 108.

Either a configuration unit or mobile handset 108 operating in system 200 and related systems below can implement similar elements to those depicted in Figure 1b for a
20 device 101, such as having a processor, memory, data bus, radio, storage, etc. The storage and memory of a configuration unit or mobile handset 108 could record an operating system to manage the hardware devices for mobile handset 108 or a configuration unit, and a software or a firmware application 101b depicted in Figure 1a and Figure 3 could conduct the message flows for unit 108 herein. In other words, for exemplary embodiments using a
25 configuration unit operating instead of the depicted "mobile handset" 108, the configuration unit could also include a configuration application 108b, and an NFC radio 108f, etc. as depicted below for mobile handset 108.

In another embodiment for Figure 2a, tag 101s could comprise an NFC tag such as a tag compatible with the NFC Forum standards for type 1 through type 5 tags (and
30 subsequent or related standards). The NFC technology could also be NFC-A, NFC-B, or NFC-V, or subsequent standards. For these exemplary embodiments where tag 101s comprises an NFC tag, then mobile handset 108 could use an NFC reader instead of camera 108r, and the NFC reader could comprise an NFC radio 108f as depicted and described in connection with Figure 3 below. Mobile handset 101 could utilize an NFC chip and
35 antenna, where the NFC chip operates in read mode with tag 101s. In exemplary embodiments, discovery server 110 can include or be associated with a discover server

database 110a. Database 110a could record information about a plurality of devices 101, such as ID.device 101b, and ID-token.device 206, and a dataset config-provisioning.ID.device 212, which will be described in further detail below for Figure 2a.

At step 201 in Figure 2a, device 101 can be in a powered off state. In alternative
5 embodiments, device 101 could be powered on at step 201, although power for device 101 is not required for embodiments with the steps and message flows outlined in Figure 2a. At step 202, mobile handset 108 could be powered on and placed in proximity of device 101. Mobile handset 108 could be operated by a device user 101a or a configuration user 108a at step 202. At step 203, device user 101a or configuration user 108a could use
10 mobile handset 108 to conduct a “read” step 203, where mobile handset 108 can (i) take a picture of a tag 101s for device 101, or (ii) read a bar code, QR code, or similar code for device 101. As mentioned above, tag 101s could comprise a bar code, QR code, or other encoding of data for device 101, including a serial number. For the alternative embodiment where (i) mobile handset 108 uses an NFC reader and (i) tag 101s comprises an NFC tag,
15 the mobile handset 108 could conduct an NFC read of tag 101s for step 203 and step 204.

For some exemplary embodiments, mobile handset 108 could receive data for the tag 101s (such as URL-DS 205, ID-token.device 206, and tag value 101w) via a packet or message received over a local wireless link between mobile handset 108 and device 101. The local wireless link could comprise any of the example wireless links of Wi-Fi,
20 Bluetooth, and near field communications, and other possibilities exist as well. With Wi-Fi networks, device 101 could periodically transmit data for tag 101s as a broadcast message, and mobile handset 108 could listen for broadcast messages in order to read the tag 101s.

At step 204, mobile handset 108 can receive data from tag 101s, such as a response, where the response can consist of a uniform resource locator (URL) for discovery server
25 110, URL-DS 205, an ID-token.device 206, and a tag value 101w. URL-DS 205 could include a domain name for discovery server 110 and parameters such as the use of HTTP or HTTPS and a destination TCP or UDP port number for sending packets to discovery server 110. ID-token.device 206 could be a token for ID.device 101b, or a value or unique pseudo-random number associated with device 101 and/or ID.device 101b. ID-
30 token.device 206 could be a value that obfuscates ID.device 101b but can preferably uniquely map to ID.device 101b. Through the use of ID-token.device 206 that is readable by mobile handset 108, then the actual value for ID.device 101b can remain confidential at step 204. Tag value 101w in response can include data associated with tag 101s and device 101, such as versions of software supported, product validity or expiration dates,
35 cryptographic parameters supported, or a subset of config-provisioning.ID.device 212. In an exemplary embodiment, tag value 101w could comprise some values of config-

provisioning.ID.device 212 upon manufacturing of device 101, which could be subsequently updated at a later date in discovery server DB 110a before installation or operation of device 101.

At step 207, mobile handset 108 could process the data received in response 204.
5 For embodiments where tag 101s is an image such as a bar code or QR code, at step 207 mobile handset 108 could process the image and extract the data or values for ID-token.device 206, URL-DS 205, and tag value 101w. Continuing at step 207, mobile handset 108 could connect with an access network 126 in order to call the URL in URL-DS 205, which could be an HTTPS/TLS session in exemplary embodiments. In exemplary
10 embodiments, URL-DS 205 is combined with ID-token.device 206, such that the HTTPS session is a unique request to discovery server 110 for each device 101. At step 207, mobile handset 108 could perform additional checks and steps in preparation for communication with device 101, such as verifying data in tag value 101w is compatible with software in mobile handset 108, including configuration software 108b.

15 In message flow 208, mobile handset 108 can establish a secure HTTPS session with discovery server 110 using the URL-DS 205, and the secure session could comprise a transport layer security (TLS) session such as TLS version 1.2, TLS version 1.3, or similar and subsequent standards. As part of the setup of message flow 208, mobile handset 108 can receive a certificate cert.DS 110b from discovery server 110. Cert.DS 110b could
20 comprise an X.509 certificate similar to the certificate depicted in Figure 4d below, and include a signed public key for discovery server 110. In exemplary embodiments, mobile handset 108 verifies the certificate cert.DS 110b using the public key in certificate cert.CA1 115. The verification of signatures using public keys in certificates is similar to the steps for verifying signatures outlined below in step 221 for Figure 2b. Although not
25 depicted in Figure 2a, in exemplary embodiments mobile handset 108 can also authenticate with discovery server 110, such as a configuration user 108a associated with mobile handset 108 entering user credentials in a web site for discovery server 110. Or mobile handset 108 could send discovery server 110 a certificate for mobile handset 108, which the discovery server 110 could verify using a step 221 in Figure 2b.

30 In message 209 and after the secure HTTPS/TLS session setup in message flow 208, mobile handset 108 can send discovery server 110 ID-token.device 206 and tag value 101w. Mobile handset could alternatively send discovery server 110 a subset of the data for ID-token.device 206 and/or tag value 101w. At step 210, discovery server 110 can use data in ID-token.device 206 to query discovery server database 110a for the values or data
35 sets ID.device 101b and config-provisioning.ID.device 212. Discovery server database 110a may comprise a database with tables to track configuration information for a plurality

of devices 101. In exemplary embodiments, Config-provisioning.ID.device 212 contains data used by mobile handset 108 for the configuration of device 101 and SE 102. Exemplary data depicted for config-provisioning.ID.device 212 in Figure 2a includes configuration application “configuration application” 108b, software parameters 108c, version number 108d, ID.Authentication Server 111a, URL-authentication Server 111b, cert.AS 111c, auth. parameters 111d. Additional or other data for mobile handset 108 and device 101 could be included in discovery server database 110a and values for config-provisioning.ID.device 212.

Configuration application 108b can specify the name or URL of a software application for mobile handset 108 to utilize in order to configure device 101 and SE 102. Mobile handset 108 could utilize the name or URL to fetch the software for configuration application 108b. Configuration application 108b could be a public or private application for the operating system of mobile handset 108. For embodiments where mobile handset 108 utilizes an Android-based operating system, configuration application 108b could be downloaded from the Google Play store or a private web site associated with configuration system 114. For embodiments where mobile handset 108 utilizes an Apple® / IOS –based operating system, configuration application 108 could be distributed through a developer enterprise program or downloaded through iTunes®. Other possibilities exist as well without departing from the scope of the present invention for a mobile handset 108 to download a configuration application 108b, where the name of configuration application 108b is received by a discovery server 110.

As depicted in Figure 2a, config-provisioning.ID.device 212 sent to mobile handset 108 may also include software parameters 108c and version number 108d. Software parameters 108c may include values and data for mobile handset 108 to utilize configuration application 108b and communicate with device 101 using an NFC radio 108f. Exemplary values include parameters 108c for mobile handset 108 to conduct a configuration step 103, such as, but not limited to, (i) cryptographic algorithms to utilize, (ii) ECC curve names and key lengths and associated data, (iii) selection and parameters for a radio 101z (below) in mobile handset 108 to communicate with device 101, including the selection of one of multiple possible radios in mobile handset 108, (iv) addresses, names, and port numbers to utilize with IP network 128, (iv) timers and retry counts for communication with device 101 or configuration server 112 or authentication server 111, (v) parameters for configuration system 114, (vi) parameters for communicating with device 101, etc. Version number 108d for config-provisioning.ID.device 212 may specify a version number of configuration application 108b to utilize, or a list or dataset of version

numbers of standards supported, such as IP version of IP network 128, NFC type or version for an NFC radio 108f, TLS version supported, etc.

As depicted in Figure 2a, config-provisioning.ID.device 212 sent to mobile handset 108 may also include ID.authentication Server 111a, URL-authentication Server 111b, cert.AS 111c, auth. parameters 111d. ID.Authentication Server 111a could comprise an identity or name for authentication server 111, such as a domain name or multiple domain names for multiple servers. URL-Authentication Server 111b could comprise a URL for configuration application 108b to call in order to authenticate with authentication server 111. In some exemplary embodiments, URL-authentication server 111b includes either ID.Device 101b or ID-token.Device 206, so that when mobile handset 108 using configuration application 108b calls URL-authentication Server 111b, the URL includes a unique string that identifies device 101.

In another exemplary embodiment, mobile handset 108 can (a) call URL-authentication server 111b without an identity for device 101, and subsequently (b) send either ID.Device 101b or ID-token.Device 206. Cert.AS 111c could comprise a certificate for authentication server 111 in a form similar to the exemplary certificate depicted below in Figure 4d, where cert.AS 111c could include a public key for certificate server 111 and a signature of a certificate authority. In exemplary embodiments, a certificate for a certificate authority associated with cert.AS 111c is sent with cert.AS 111c. Authentication parameters 111d can include parameters for configuration application 108b to utilize with authentication server 111, such as, but not limited to, hash or message digest algorithms, digital signature algorithms, padding schemes, key lengths, nonce or challenge lengths, timers, and retry counts, encoding rules, etc.

The above fields and values described for config-provisioning.ID.device 212 and ID.device 101b could be sent from discovery server 110 to mobile handset 108 in a response 211 as depicted in Figure 2a. In an exemplary embodiment, discovery server 110 can send mobile handset the full configuration application 108b with the response 211 instead of a name or URL for configuration application 108b, if the current, updated configuration application 108b is not already downloaded by mobile handset 108. In another exemplary embodiment, mobile handset 108 can download configuration application 108b in a step 213 as depicted in Figure 2a, if mobile handset 108 has not already installed the software for configuration application 108b after receiving response 211. For example, response 211 could include a version number 108d that is higher or different than initially utilized by mobile handset 108 and/or an initial configuration application 108b, and consequently the higher version number 108d could indicate to mobile handset 108 that a different configuration application 108b would be required. In

exemplary embodiments, configuration application 108b can include a hash signature similar that is created using a signature creation 220 step as depicted in Figure 2b below. Step 213 may also include mobile handset 108 launching the configuration application 108b and applying a subset of software parameters 108c or tag-value 101w. If any of the
5 steps depicted in Figure 2a fail, error codes or messages could be displayed through mobile handset 108 to configuration user 108a.

After successful launch of configuration application 108b, mobile handset 108 can connect with authentication server 111. Secure session setup 214 could comprise a secure
10 HTTPS session with authentication server 111 using the URL-authentication server 111b, and the secure session could comprise a transport layer security (TLS) session such as TLS version 1.2, TLS version 1.3, or similar and subsequent standards. Other networking standards besides HTTPS and TLS could be utilized as well for secure session setup 214, such as an IPsec tunnel. As part of the setup of secure session setup 214, mobile handset
15 108 can receive a certificate cert.AS 111c from authentication server 111, if mobile handset 108 had not already received cert.AS 111c from discovery server 110. Mobile handset 108 could verify cert.AS 111c after receiving the certificate, such as verifying a signature from a certificate authority and parent certificates of cert.AS 111c as well.

In message 215, mobile handset 108 transmits authentication information to authentication server 111, and authentication server 111 verifies the authentication
20 information in a step 217. The authentication information transmitted in a message 215 could comprise information to authenticate (i) configuration user 108a, (ii) mobile handset 108, or (iii) both configuration user 108a and mobile handset 108. The authentication information in message 215 may comprise a configuration user 108a entering identification and credential information in a touch screen of mobile handset 108, and the information is
25 passed by mobile handset to authentication server 111. In exemplary embodiments, configuration user 108a provides biometric data such as a fingerprint, an image of the user's face, or similar data, where mobile handset 108 transmits the data through secure session setup 214 to authentication server 111, and the authentication server 111 verifies the biometric data is associated to configuration user 108a. Message 215 may also include
30 the verification of mobile handset 108 with authentication server 111.

In exemplary embodiments, mobile handset 108 records a certificate for mobile handset 108, where the certificate could be similar to the one depicted in Figure 4d below. The certificate for mobile handset 108 includes a public key and a signature by a certificate authority. In an authentication step 217, authentication server can use the information in
35 message 215 to verify the identity of a configuration user 108 and/or mobile handset 108. In embodiments where mobile handset 108 uses a device certificate for mobile handset

108, authentication server 111 could verify that mobile handset 108 also records the private key, by sending a challenge and verifying a signature from mobile handset 108 using the private key. Message 215 and authentication 217 could be conducted using authentication parameters 111d, which could be previously included in Config-provisioning.ID.device
5 212.

In step 216a of Figure 2a, mobile handset 108 may also authenticate authentication server 111, using cert.AS 111c. In step 216b, configuration user 108b can enter device owner 122 information into mobile handset 108, such as an owner name, a DNS name associated with owner 122, or a code or other value to identify owner 122. Device owner
10 122 information could be automatically recorded for configuration application 108b before a configuration step 103 from Figure 1a. The value or information associated with device owner 122 may also be acquired from monitored unit 106, for embodiments where device owner 122 also owns or is responsible for monitored unit 106. The value associate with device owner 122 could be ID.owner 122a. In another exemplary embodiment, device
15 owner 122 information could be recorded in mobile handset 108 earlier than a step 216b, and the information could be fetched from memory in a step 216b. A step 216b can function to associate device 101 with device owner 122 for a configuration system 114 and/or reporting system 120. In exemplary embodiments, millions of devices 101 could be distributed out to thousands or more device owners 122, and configuration system 114 and
20 reporting system 120, as well as device owner 122, may need to know which devices 101 have been configured and/or installed with monitored units 106.

In message 218, mobile handset 108 can send a first list of identities to authentication server 111 through the secure session setup in 214. The list of identities can include an ID.device 101b, ID.MH 108e, and ID.owner 122a, as depicted in Figure 2a. In
25 steps 219, authentication server 111 can process the list of identities received in message 218. A first sub-step 219a could comprise authentication server 111 verifying that mobile handset 108 with configuration user 108a is authorized to configure device 101. In other words, the prior step 217 could be an initial step to verify identities, and then the subsequent step 219a could be verifying that the authenticated identities have privileges or
30 authorization to conduct a configuration step 103. As depicted in Figure 2a, authentication server 111 may need to check or communicate with device owner 122 in order to confirm that mobile handset 108 with configuration user 108a is authorized to conduct configuration step 103 on device 101.

Authentication server 111 can utilize the ID.owner 122a received in message 218 in
35 order to identify and communicate with the correct device owner 122. Although not depicted in Figure 2a, authentication server 111 could utilize a mutually authenticated

secure channel such as TLS v3 with device owner 122 in a step 219a and 219b. In some exemplary embodiments, a step 219a could be optionally omitted, for example if device owner 122 is also device user 101a, such as a home owner installing a device 101 in their home. In these embodiments, an authentication step 217 may be sufficient to allow mobile handset 108 to conduct the configuration step 103. Other possibilities exist as well for an authentication server to authenticate a mobile handset 108 and/or configuration user 108a without departing from the scope of the present invention.

In step 219b, authentication server 111 can query device owner 122 for a certificate for secure processing environment 102 (SE 102), and the certificate could comprise cert0.SE 102k and cert0.SE 102k', where cert0.SE 102k can record PK0.SE 102n and cert0.SE 102k' can record PK0.SE 102n'. Note that in some exemplary embodiments cert0.SE 102k can record two public keys for device 101 of both PK0.SE 102n and PK0.SE 102n'. The query could be made using ID.device 101b, which was previously received in message 218. Cert0.SE 102k could include ID.SE 102m, or ID.SE 102m could be received by authentication server 111 separately from cert0.SE 102k. In a different exemplary embodiment, the query in a step 219b could utilize ID-token.device 206, which could be (i) obtained by mobile handset 108 in read 203, and (ii) sent by mobile handset 108 in message 218. Cert0.SE 102k could also be a certificate for device 101 in some exemplary embodiments, where device 101 does not use separate certificates for the device and secure processing environment.

If owner 122 does not record the certificate cert0.SE 102k, then for step 219a owner 122 could point authentication server 111 to another server or location for the certificate cert0.SE 102k, such as possibly device manufacturer 101x. In some exemplary embodiments, (i) a configuration system 114 could query another server associated with device 101 in order to obtain public key PK0.SE 102n, and (ii) the use of a cert0.SE 102k for recording the public key PK0.SE 102n is not required. Authentication server 111 could receive cert0.SE 102k in a step 219b. In another exemplary embodiment, authentication server 111 could record a plurality of certificates cert0.SE 102k for all devices 101 and in this embodiment then step 219b could represent querying a locally stored cert0.SE 102k with authentication server 111.

In step 220a, authentication server 111 can (i) create a random number random1.AS 223, and (ii) create a digital signature 224. The sub-steps for creating a digital signature are depicted in Figure 2b below for a signature creation step 220. Exemplary details for authentication server 111 to create specifically create digital signature 224 in a step 220a are depicted and described in connection with Figure 2b below.

Authentication server 111 can also conduct a step 220b, which comprises creating a digital signature 225 using the data received in step 219b above, such as the ID.SE 102m, where the digital signature 225 is a signature for the data of ID.SE 102m and a random1.AS 223. Step 220b is depicted in Figure 2b below, and can include the data and steps for a signature creation step 220 equivalent to the step 220a for a signature creation. Although the use of two signatures 224 and 225 are depicted in Figure 2a, where signature 224 can be used primarily by mobile handset 108 and signature 225 can be used primarily by device 101 and SE 102, the use of two separate signatures at step 220a and 220b can be optionally omitted. For example, a step 220a creating signature 224 to authenticate authentication server 111 with mobile handset 108 could be performed with a step 216a above.

As depicted in Figure 2a, authentication server 111 can send mobile handset 108 a message 222 through secure connection 214, where message 222 can include ID.Device 101b, Random1.AS 223, Signature1.AS (ID.Device 101b, Random1.AS 223) 224, and Signature2.AS (ID.SE 102m, Random 1.AS 223) 225. Although not depicted in Figure 2a, a message 222 could also include cert0.SE 102k. Random1.AS 223 may comprise a pseudo-random number generated by authentication server 111 for message 222 and signature 224. Random1.AS 223 and other random numbers utilized by systems 100, 200, 300, and subsequent figures may preferably be unique, or with enough bits and information entropy that collisions or re-use would be sufficiently rare enough to be insignificant for the overall security of a system 100. Signature1.AS 224 is described in Figure 2b below. Signature2.AS 225 was described in the paragraph above and also depicted in Figure 2b below. Authentication server 111 could utilize authentication parameters 111d in the creation of data for message 222. Although not depicted in Figure 2a, in exemplary embodiments, authentication server 111 could also include authentication parameters 111d for the data in signature1.AS 224 and signature2.AS 225, and in this manner authentication parameters 111d would be signed by authentication server 111.

In exemplary embodiments, ID.SE 102m is not sent either in encrypted format or as plaintext to mobile handset 108 from either (i) discovery server 110 or (ii) authentication server 111, or (iii) from other servers in a system 100. By not sending ID.SE 102m to mobile handset 108, ID.SE 102m can remain confidential, such that if mobile handset 108 is insecure and exposed to third parties, including hackers, ID.SE 102m can still remain confidential since it is not passed to mobile handset 108. Signature2.AS 225 (ID.SE 102m, random1.AS 223) is depicted in Figure 2b below, and can comprise a signature by authentication server 111 over random1.AS 223 and ID.SE 102m. ID.SE 102m can comprise a device value, where the device value is a unique number or string associated

with device 101. The device value can be recorded in SE 102 before a configuration step 103 such as during manufacturing of SE 102.

At step 221a in Figure 2a, mobile handset 108 can verify signature1.AS 224, where signature1.AS 224 can be over the values ID.device 101b and random1.AS 223. The sub-steps for verifying a digital signature are depicted in Figure 2b below for a signature verification step 221, and the details for mobile handset 108 to verify digital signature 224 in a step 221a are depicted and described in connection with Figure 2b below. ID.device 101b in message 222 can be useful to track that message 222 is properly received routed by authentication server 111 since both authentication server 111 and mobile handset 108 may connect with multiple different devices 101 over time. In step 221a, mobile handset 108 can take steps to verify that random1.AS 223 is unique and not reused by authentication server 111, such as checking random1.AS 223 with configuration system 114. In exemplary embodiments, authentication server 111 can create a signature 224 over a random number issued by mobile handset 108 instead of a random number generated by authentication server 111, which could be conducted in setup of secure session 214. Or, in preferred exemplary embodiments, authentication server 111 creates a signature 224 for both (i) a random number from mobile handset 108 in a step 216a and (ii) a random1.AS 223 in a step 220b. After conclusion of step 221a depicted in Figure 2a, mobile handset 108, device 101, and the elements depicted in Figure 2a could continue with the message flows and steps depicted in Figure 3 below.

In an alternative exemplary embodiment to that depicted in Figure 2a, discovery server 110 could be optionally omitted, and data within tag 101s or tag value 101w could specify authentication server 111 as well as data for Config-provisioning.ID.device 212. However, the use of a discovery server 110 can be preferred for some exemplary embodiments since a discovery server 110 could provide increased flexibility over the lifetime of device 101, which could be a decade or longer, since parameters Config-provisioning.ID.device 212 and related data could change over time. Including all data for config-provisioning.ID.device 212 in tag value 101w or tag 101s may reduce flexibility for changing or updating the data over time, such as specifying an updated configuration application 108b, or other updated data for configuration system 114 over time.

Figure 2b

Figure 2b is a flow chart illustrating exemplary steps for creating and verifying a digital signature using PKI keys, parameters, and data input, in accordance with exemplary embodiments. The processes and operations, described below with respect to all of the logic flow diagrams and flow charts may include the manipulation of signals by a processor

and the maintenance of these signals within data structures resident in one or more memory storage devices. For the purposes of this discussion, a process can be generally conceived to be a sequence of computer-executed steps leading to a desired result.

These steps usually require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is convention for those skilled in the art to refer to representations of these signals as bits, bytes, words, information, elements, symbols, characters, numbers, points, data, entries, objects, images, files, or the like. It should be kept in mind, however, that these and similar terms are associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as listing, creating, adding, calculating, comparing, moving, receiving, determining, configuring, identifying, populating, loading, performing, executing, storing etc. that are often associated with manual operations performed by a human operator. The operations described herein can be machine operations performed in conjunction with various input provided by a human operator or user that interacts with the device, wherein one function of the device can be a computer.

In addition, it should be understood that the programs, processes, methods, etc. described herein are not related or limited to any particular computer or apparatus. Rather, various types of general purpose machines may be used with the following process in accordance with the teachings described herein.

The present invention may comprise a computer program or hardware or a combination thereof which embodies the functions described herein and illustrated in the appended flow charts. However, it should be apparent that there could be many different ways of implementing the invention in computer programming or hardware design, and the invention should not be construed as limited to any one set of computer program instructions.

Further, a skilled programmer would be able to write such a computer program or identify the appropriate hardware circuits to implement the disclosed invention without difficulty based on the flow charts and associated description in the application text, for example. Therefore, disclosure of a particular set of program code instructions or detailed hardware devices is not considered necessary for an adequate understanding of how to make and use the invention. The inventive functionality of the claimed computer

implemented processes will be explained in more detail in the following description in conjunction with the remaining Figures illustrating other process flows.

Further, certain steps in the processes or process flow described in all of the logic flow diagrams below must naturally precede others for the present invention to function as described. However, the present invention is not limited to the order of the steps described if such order or sequence does not alter the functionality of the present invention. That is, it is recognized that some steps may be performed before, after, or in parallel other steps without departing from the scope and spirit of the present invention.

The processes, operations, and steps performed by the hardware and software described in this document usually include the manipulation of signals by a CPU or remote server and the maintenance of these signals within data structures resident in one or more of the local or remote memory storage devices. Such data structures impose a physical organization upon the collection of data stored within a memory storage device and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

In Figure 2b, signature creation 220 can comprise a step using the sub-steps of obtaining a message to sign, calculating a message digest 230, using a private key 226, using a signature algorithm 227, inputting parameters 111d, and calculating a resulting signature 224. The steps and substeps depicted for authentication server creating signature 224 in Figure 2b can also be applied for the subsequent signatures depicted, including signature creation 220b through 220g.

Signature algorithm 227 and signature verification algorithm 228 could comprise an RSA-based digital signature algorithm (DSA) or an ECC based elliptic curve digital signature algorithm (ECDSA), and other possibilities exist as well for signature algorithm 227 and signature verification algorithm 228 without departing from the scope of the present invention. When using a DSA or ECDSA algorithm in non-deterministic mode for a signature creation 220, a value of “k” or “r”, which could comprise a random number can be associated with the digital signature 224. When using a DSA or ECDSA in deterministic mode, such as specified in IETF RFC 6979 and titled “Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)”, which are hereby incorporated by reference, then the requirement for a separately transmitted random number with digital signature 224 (such as value “k” or “r”) can be optionally omitted, such that “r” can be deterministically calculated based on the message to sign. In exemplary embodiments, device 101 and servers such as authentication sever 111 can utilize deterministic ECDSA without also sending a random number along with a

digital signature, although the value “r” from the deterministic mode could be sent with the digital signature 224. In other words, a value can be sent with the digital signature 224 that is deterministically derived and associated with the message to sign. In other exemplary embodiments, a random number can be generated a derived value for the random number such as “r” sent with digital signature 224.

For a signature creation 220a step, the exemplary message to sign comprises ID.device 101b and random I.AS 223. The message to sign values can be transmitted to the verifying party, such as shown for message 222 in Figure 2a. The message to sign values can be input into a message digest algorithm 230, which could comprise a standard algorithm such as SHA-256, SHA-3, or similar algorithms. The output of message digest algorithm 230 can be input along with parameters 111d and private key 226 into signature algorithm 227. Parameters 111d can specify encoding rules, padding, key lengths, selected algorithms, curve names, and other values or fields necessary to utilize a signature algorithm 227. Both a signature creation step 220 and a signature verification step 221 use the same or equivalent values for parameters 111d. Private key 226 comprises a private key for authentication server 111. For the additional signature creation 220 instances depicted in Figure 2b, such as signature creation 220b through 220g, the element or node creating the equivalent signature as signature 224 would utilize the private key of the element or node, such as SE 102 using a private key SK0.SE 102y’ in a signature creation instance 220c. Although not depicted in Figure 2b, a random number for values such as “k” and “r” for ECDSA and related signatures could be input as well into a signature algorithm 227. The output of a signature algorithm 227 can be a signature 224, which can be transmitted to another node for verification.

Signature verification 221 can comprise a step using the sub-steps of obtaining a message to verify, calculating a message digest 230, using a public key 228, using a signature verification algorithm 228, inputting parameters 111d and signature 224, and determining a pass/fail 229. If the signature 224 received matches a calculated signature with the public key 228 and message to sign, then the received signature 224 passes or is validated or verified. If the signature 224 does not match the calculated signature in a step 221, then the signature 224 is considered to fail or not be verified. The steps and substeps depicted for mobile handset 108 verifying signature 224 in Figure 2b can also be applied for the subsequent signature verifications depicted, including signature verification 221b through 221g.

Figure 3

Figure 3 is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a mobile handset and a device, in accordance with exemplary embodiments. Before initiating steps and message flows depicted in Figure 3, mobile handset 108, device 101, and the other elements depicted for system 300 in Figure 3 may previously complete exemplary message flows and steps depicted in Figure 2a above. System 300 can include a configuration server 112, authentication server 111, mobile handset 108, and device 101. Mobile handset 108 can communicate with authentication server 111 and configuration server 112 via IP network 128, as depicted in Figure 2a. For system 300, configuration server 112 and authentication server 111 may establish a secure session setup 301, which could comprise establishing a secure communications link between the two servers using protocols such as TLS, IPSec, a virtual private network (VPN), a secure shell (SSH), or similar networking, transport, or application layer technologies in order to establish secure communications between configuration server 112 and authentication server 111. Secure session setup 301 can be equivalent or similar to secure session setup 214 depicted in Figure 2a above.

Note that for some exemplary embodiments, a mobile handset 108 could be omitted, and device 101 and configuration system 114 (with authentication server 111 and configuration server 114) could communicate directly with device 101. For example, device 101 could use a radio such as a 4G or 5G radio or a Wi-Fi radio to communicate with configuration system 114 without using a mobile handset 108, and data session 303 could comprise a data session that includes WAN connectivity. For these embodiments, then the exemplary depicted messages sent or transmitted by configuration system of message 222 (from Fig. 2a), 312, etc. could be transmitted to device 101 instead of mobile handset 108. Further, messages sent or transmitted by device 101 of 307 and 314 could be sent to configuration system instead of mobile handset 108. However, for embodiments where device 101 does not have a configured WAN radio, then a system equivalent to system 200 and 300 could be preferred. Also, note that a different device than a mobile handset 108 could be used for establishing communication between device 101 and configuration system 114, and a Wi-Fi router or another node on a local wireless network could provide the same functionality of mobile handset 108 (where tag 101s could be transmitted via the local wireless network).

Mobile handset 108 in system 300 can operate a running configuration application 108b and an NFC radio 108f. Configuration application 108b may previously be downloaded in a step 213 as depicted in Figure 2a and subsequently launched or activated for operation in a system 300. In exemplary embodiments, configuration application 108b may be (i) included or distributed along with an operating system running on mobile

handset 108b or (ii) acquired at a different time than a step 213, and thus a separate download of configuration application 108b. NFC radio 108f may comprise a radio equivalent or compatible with radio 102c depicted for SE 102 in Figure 1b above. Although radio 108f is depicted in Figure 3 as a “Near Field Communications” (NFC) radio, in alternative embodiments a different radio than NFC could be utilized, such as a radio operating at ISM bands of 900 MHz, 2.4 GHz, 5.8 GHz, or other radio frequencies allowing unlicensed access at low powers such as less than 1 watt of transmitted radio power. As depicted in Figure 3, device 101 can include a secure processing environment (or “secure enclave” or “secure element”) SE 102 which may also have a NFC radio 102c, thereby enabling communication at short range between mobile handset 108 and device 101. In some exemplary embodiments, SE 102 may be connected with NFC radio 102c and NFC radio 102c may be external to SE 102, but in preferred exemplary embodiments NFC radio 102c may be inside device 101.

At step 302 in Figure 3, device 101 may power on from an unpowered or “deep sleep” state. Power could be provided from a battery or converted “wall power”. Device 101 could load portions of OS 101e from storage 101f into RAM 101d. SE 102 could load SE bootload firmware 102t and SE boot configuration 102v, and then SE firmware from SE memory 102x. In exemplary embodiments where SE 102 has sufficient storage memory to record SE firmware and an operating system for SE 102 in SE flash memory 102d, then SE firmware could be loaded from SE flash memory 102d instead of SE memory 102x. In exemplary embodiments, SE memory 102x within device memory 101f is encrypted with symmetric key 102z.

Mobile handset 108 and SE 102 can establish an NFC session setup 303 for peers and “peer-to-peer mode”. Either Mobile handset 108 or SE 102 could initiate the session once the two sides are sufficiently close for the radios to communicate. In exemplary embodiments, a screen on mobile handset 108 instructs configuration user 108a to place the mobile handset near device 101 or next to a target location on device 101 that would cover an antenna for SE radio 102c. In a first exemplary embodiment, NFC radio 108f in mobile handset 108 is the transceiver that initiates polling and in the “initiator” of NFC session setup 303. For this first exemplary embodiment SE radio 102c in SE 102 comprises the target and initially listens for polling from NFC radio 108f. In a second exemplary embodiment, SE radio 102c for SE 102 is the transceiver that initiates polling, and is the “initiator” of NFC session setup 303. After negotiating polling, NFC session setup 303 can utilize protocols for either NFC-A or NFC-F to establish a bi-directional communications channel, or subsequent and related versions of these standards. NFC-A is specified in NFC

Forum-TS Digital Protocol-1.0, which is hereby incorporated by reference and provides a communications channels at baud rates of 106 kbps.

In exemplary embodiments NFC-F is utilized to provide baud rates higher than 106 kbps, such as 212 or 424 kbps. The air interface for NFC session setup 303 could comprise
5 a session using ISO/IEC 18092 / ECMA-340, "Near Field Communication Interface and Protocol-1" and ISO/IEC 21481 / ECMA-352, "Near Field Communication Interface and Protocol-2 (NFCIP-2)". The air interface for NFC session setup 303 could also comprise subsequent or related versions to these standards. Data between mobile handset 108 and SE 102 could be transferred using NFC Data Exchange Format (NDEF) and support a
10 Multipurpose Internet Mail Extensions (MIME) typed object or file transferred from mobile handset 108 to SE 102. In other words, subsequent messages and files transferred using NFC radios could utilize NDEF and MIME objects, although other possibilities exist as well for the NFC standards utilized without departing from the scope of the present invention. NFC session setup 303 could also implement additional air-interface security,
15 such as ECMA-409 2nd edition - NFC-SEC-02: NFC-SEC and related standards, where the air interface is encrypted using AES and Diffie-Hellman key exchanges. In exemplary embodiments, NFC session setup 303 utilizes standard ECMA-352 in order to select ECMA-340, ISO/IEC 14443 and ISO/IEC 15693 as a communication mode between radio 108f and radio 102c.

20 At step 304 in Figure 3, after successful setup of NFC session setup 303, configuration application 108b in mobile handset 108 could display to configuration user 108a that a communication link with SE 102 has been established. The display could be notification that authentication data can be transferred with SE 102, such as the data through the physical layer and data link layer of NFC radio 108f is enabled. Step 304
25 could also display error conditions or codes to configuration user 108a, for cases where NFC session setup 303 fails for exemplary reasons such as no NFC radio 102c detected, incompatible radio protocols, too many collisions, too weak an RF signal or RSSI, etc. Note that in an exemplary embodiment, secure session setup 214 may be temporarily disabled or unavailable (such as mobile handset 108 moves outside of range of a mobile
30 network providing connectivity to authentication server 111). In this case, the values subsequently transmitted in message 305 and received in message 307 could be stored in mobile handset 108 until they can be transmitted from mobile handset 108. In this manner, if device 101 is located in a place without an access network 126, such as in a basement of a building where nearby land mobile networks cannot reach, then mobile handset 108 can
35 take subsequent steps such as step 305 while mobile handset 108 is in an offline mode for access network 126.

In message 305, mobile handset 108 can send SE 102 data comprising certificate cert.AS 111c, random number Random 1.AS 223, authentication parameters Auth.Params.111d, and signature Signature2.AS (ID.SE, Auth.Params.111d, Random 1.AS) 225. These values could be previously received and recorded by mobile handset 108 during the prior communication depicted and described in connection with Figure 2a above. Although not depicted in Figure 3, mobile handset 108 could also send other exemplary data in a message 305 than that depicted, including parent certificates for cert.AS 111c, version number 108d, ID.device 101b, and/or ID.mobile-handset 108e. In exemplary embodiments, message 305 is transmitted without receipt of network, transport, or application layer data from SE 102, and SE 102 remains “quiet” or does not communicate data at the network, transport, or application layer with mobile handset 108 until after successfully receiving a message 305. In other words, although SE 102 may communicate at the data-link layer, such as transmitting values related to polling the NFC link, encryption of the radio air-interface for session 303, SE 102 does not respond to mobile handset 108 with application data until after message 305 has been properly received and verified. In this manner, SE 102 would (i) remain silent or idle to a mobile handset 108 or other NFC radios without receiving a verified message 305 and thus (ii) resistant to tampering or probing by radios or handsets that have not established an authenticated session with authentication server 111.

Although not depicted in Figure 3, in an exemplary embodiment a signature2.AS 225 could also be over an initial random number for SE 102 or device 101, which would be different than, in addition to, or in replace of the value random1.AS 223 for a signature 2.AS 225. The initial random number for SE 102 could be (i) recorded in the tag 101s depicted and described in Figure 2a, or (ii) recorded by owner 122 such that authentication server 111 could query owner 122 in a step 219a as depicted in Figure 2a above, or (iii) recorded with date in Config-provisioning.ID.device 212. The initial random number for SE 102 could also be recorded in memory 102d or memory 102j for SE 102 and created by a device manufacturer 101x. In this manner, the initial random number for SE 102 can be generated outside authentication server 111 and included in a signature2.AS 225. Security can be increased since authentication server 111 can be required to sign a number that was both (i) not generated by authentication server 111 and (ii) recorded in device 101 before a configuration step 103 or step 203.

In step 306a, SE 102 can receive the message 305 via the NFC session 303 and record the data in RAM 102b or storage 102d. In step 306a, SE 102 can also verify cert.AS 111c using either (i) cert.CA1 115 directly, and/or (ii) a commonly shared certificate cert.CA.root 109 shared between SE 102 and authentication server 111. In other words, in

exemplary embodiments and before a configuration step 103, SE 102 records in nonvolatile memory either (i) cert.CA1 115 or (ii) cert.CA.root 109, where a parent certificate for cert.AS 111c can be checked with cert.CA.root 109. As described in connection with Figure 2b above, SE 102 could verify the certificate for cert.AS 111c using a signature verification step 221.

After a step 306a, SE 102 can conduct a step 221b depicted in Figure 2b in order to verify signature 225. In exemplary embodiments, signature 225 can be over ID.SE 102m, randomI.AS 223, and authentication parameters 111d, where (i) randomI.AS 223 and authentication parameters 111d are received by SE 102 in a message 305 above and (ii) ID.SE 102m may optionally be not transmitted by mobile handset 108 but rather recorded by SE 102 in nonvolatile memory before configuration step 103. In this manner, SE 102 can receive a signature 225 for data (a) that has not been transmitted by SE 102 (e.g. ID.SE 102m, and/or optionally the initial random number for SE 102 described two paragraphs above), and also (b) mobile handset 108 would need to communicate with configuration system 114 in an authenticated manner in order to receive signature 225.

For embodiments depicted in Figure 3 and Figure 2a, where ID.SE 102m is not transmitted by device 101 before receipt of a message 305 with signature 225, then ID.SE 102m within signature 225 can also function as an authentication token for device 101 and SE 102. The use of an authentication token within signature 225 is also described in the paragraph below. In some embodiments, such as those using lower security levels or where an authentication token is not available, then the use of an authentication token such as ID.SE 102m could be omitted. In another exemplary embodiment, authentication parameters 111d may be optionally omitted from signature 225, such as both (i) not being included in the “message to sign” in signature creation 220 in Figure 2b and (ii) not being included in the “message to verify” in signature verification 221 in Figure 2b. However, including authentication parameters 111d in signature 225 may increase security for SE 102.

Although the use of an ID.SE 102m is depicted as being utilized for signature 225 received from mobile handset 108 and authentication server 111 (in Figure 2a above), other data recorded by SE 102 and not previously transmitted by SE 102 to mobile handset 108 could be utilized as well for the signature 225, such as an authentication token recorded in nonvolatile memory of SE 102. A signature 225 could also be over (i) the authentication token as described in the paragraph above, (ii) the initial random number for SE 102 described four paragraphs above (which could be recorded in the tag 101s), and (iii) randomI.AS 223. In other words, in a different embodiment than that depicted in Figure 3, for a message 305 SE 102 could receive a signature 225 over the authentication token

recorded in nonvolatile memory of SE 102, where (i) SE 102 has not transmitted the authentication token, but (ii) configuration system 114 could receive the authentication token from device manufacturer 101x or device owner 122 or possibly device user 101a. By receiving a signature 225 over data recorded in nonvolatile memory of SE 102 and not transmitted by SE 102, SE 102 can be reasonable assured that mobile handset 108 has authenticated and secured communication with configuration system 114.

In step 306b, SE 102 may use a random number generator 102e from Figure 1b to create a random number random1.SE 308, which could comprise a challenge or nonce in order to authenticate configuration system 114. As depicted in Figure 3, SE 102 may then conduct a step 220c from Figure 2b in order to create a signature 309 for random1.AS 223 using the initial secret key SK0.SE 102y' as the private key for signature 309 creation. Signature 309 could also be over additional data than random1.AS 223, such as random1.SE 308, although this alternative embodiment is not depicted in Figure 3. Further, SE 102 could send separate signatures 309 such as a first signature 309 over random1.AS 223 and a second signature 309 over random1.SE 308. SE 102 may then return a message 307 though NFC session 303 to mobile handset 108, where message 307 can include random1.SE 308 and signature 309, where signature 309 can be over at least random1.AS 223. Although not depicted in Figure 3, message 307 can also optionally include an identity for device 101 and the value of PK0.SE 102n, in order for a configuration system 114 to use with subsequent communications with device 101.

At step 310, mobile handset 108 can receive message 307 and display to configuration user 108a that application-layer data has been successfully transferred with SE 102. In other words, a first display progress indicator could be shown for configuration user 108a when NFC session setup 303 was completed, and a second display progress indicator for configuration user 108a could be shown when message 307 is received. The receipt of message 307 may indicate that message 305 was successfully received and properly processed by SE 102. For embodiments where mobile handset 108 records ID.SE 102m, mobile handset 108 could request for a cert0.SE 102k' from SE 102, owner 122, or configuration system 114 before a step 310, and then authenticate signature 309 in a step 310 using the cert0.SE 102k'. However, authentication of signature 309 by mobile handset 108 is not required in some exemplary embodiments. For embodiments where mobile handset 108 does not have access to access network 126 at the physical location of device 101, then mobile handset 108 could record the data from message 307 in memory and transmit the data to authentication server 111 at a later time when communication with access network 126 was re-established, such as when a mobile network comes back into range for mobile handset 108.

After a step 310, mobile handset 108 can send authentication server 111 a message 311, where message 311 can include ID.device 101b, random1.SE 308, and signature 309 over at least random1.AS 223. Message 311 could be transmitted over a secure session 214 depicted in Figure 2a above. For embodiments where device 101 sends PK0.SE 102n in a message 307, then mobile handset 108 can also send PK0.SE 102n in a message 311. Authentication server 111 can previously maintain a mapping of ID.device 101b and ID.SE 102m based upon the prior steps in Figure 2a above, such as when sending message 222. In addition, authentication server 111 can record authentication parameters 111d for authentication steps with device 101 and SE 102. In other words, a first set of authentication parameters 111d could be used with a first set of devices 101 and a second set of authentication parameters 111d could be use with a second set of devices 101.

Authentication server 111 could also record cert0.SE 102k. Authentication server could conduct a step 221c depicted in Figure 2b in order to verify signature 309 using the PK0.SE 102n' recorded within cert0.SE 102k or cert0SE 102k'. Step 221c may comprise an authentication step for SE 102, such that an identity for SE 102 is verified with configuration system 114 using a "passed" 229 signature 309, since signature 309 was over the random number or "challenge" issued by authentication server 111 in random1.AS 223. In exemplary embodiments, random1.AS 223 is unique and with a sufficient length in order to reduce the possibility of reuse, such as an exemplary length of 32 or 64 bytes, and other possibilities exist as well.

Authentication server 111 may then utilize random1.SE 308 to create signature3.AS (random1.SE 308) 313. Signature3.AS 303 over random1.SE 308 could utilize a signature creation step 220d as depicted in Figure 2b. Signature3.AS 313 can comprise the signature by authentication server 111 for the random number or challenge issued by SE 102 in order to SE 102 to authenticate configuration system 114. Authentication server 111 can send mobile handset 108 a message 312 with signature3.AS 313, where signature3.AS 313 was created in step 220d. Mobile handset 108 can then send SE 102 a message 314, where message 314 can include the signature3.AS 313 over random1.SE 308. SE 102 can receive the message 314 over the radio 102c and verify the signature3.AS 313 using a step 221d as depicted in Figure 2b. SE 102 could input (i) the value for random1.SE 308 issued by SE 102 above into the "message to verify" in a signature verification 221 step and also input (ii) the public key recorded in certificate cert.AS 111c in order to verify signature3.AS 313. When signature3.AS 313 is verified by SE 102, then authentication server 111 and associated configuration network 114 can be authenticated with SE 102, since authentication server 111 was able to properly sign the random challenge issued by SE 102.

Upon successful verification of signature3.AS 313 in a step 221d, SE 102 can send mobile handset 108 an "OK" message in message 316 or other error code. Although not depicted in Figure 3, SE 102 could optionally send along with the "OK" a second random number random2.SE generated by random number generator 102e in SE 102. Random2.SE could be utilized by configuration system 114. In exemplary embodiments, SE 102 could also send additional configuration data for SE 102 along with the OK, such as an updated set of authentication parameters 111d, information about device 101 that is recorded by SE 102, and other possibilities exist as well for data that SE 102 sends after successfully conducting a step 221d. Mobile handset 108 can also forward the "OK" and optional additional information to authentication server 111.

After the mutual authentication depicted in Figure 3, authentication server 111 can then send a message 317 to configuration server 112 the ID.device 101b, ID.mobile-handset 108e, and random1.SE 308. In an alternative exemplary embodiment, message 317 could include random2.SE instead of random1.SE 308, where random2.SE was described above in this paragraph. Configuration server 112 may then perform a step 318 to record that (i) device 101 with ID.device 101b and (ii) mobile handset 108 with ID.mobile-handset 108e are authenticated for configuration system 114. Message 317 could also optionally include encryption or session keys for previously established secure communications with mobile handset 108, such that the encryption or session keys could be reused from a secure session 214.

As depicted in Figure 3, mobile handset 108 using configuration application 108b can conduct a step 319 to collect an identity list 320, where identity list 320 comprises a list of identities for elements of systems 100, 200, and 300 around or nearby to device 101 and monitored unit 106. Identity list 320 can include a networks available list 322. In order to conduct a step 319, mobile handset 108 may need to temporarily disconnect from an access network 126, such as a mobile phone network, in order to perform a full frequency scan of all available mobile network operators and network access technologies around device 101 in order to collect a networks available list 322. The networks available list 322 can be useful for configuration system 114 to select the best or a preferred access network 126 from the networks available list 322. As depicted in Figure 3, a networks available list 322 can also include the access technology available from a plurality of access networks 126 around device 101, such as 2G, 3G, 4G LTE, 5G, Wi-Fi, NB-IoT, LPWAN, etc. In exemplary embodiments, the networks available list 322 of surrounding wireless networks also includes an associated RF signal strength for each, such as RSSI as well as a frequency band, frequency list in MHz, or RF channel for each surrounding wireless network.

A step 319 may also comprise mobile handset 108 collecting a list of identities for monitored units 106 and transducers 105 for device 101, and include the identities in the identity list 320. In exemplary embodiments, a step 319 in Figure 3 may comprise mobile handset 108 scanning for a bar code or QR code on each of the monitored units 106 and transducers 105 in order to collect an identity for each. Mobile handset 108 could also or alternatively take a picture and record the picture for each monitored unit 106 and transducer 105. The picture for monitored unit 106 and transducer 105 could be processed by mobile handset 108 or configuration system 114 in order to identify monitored unit 106 and transducer 105. The identity of owner 122 for device 101 could be entered by the user in a screen or previously acquired by mobile handset 108, such as with a set of configuration parameters config-provisioning.ID.device 212.

Further, in a step 319, mobile handset 108 can preferably obtain (i) geographical coordinates for a data field location.device 325 or (ii) estimated geographical coordinates for the physical location of device 101. In exemplary embodiments, mobile handset 108 can have a GPS or similar receiver for outdoor location, or use Wi-Fi or Bluetooth for approximate indoor locations, and record the location in location.device 325. Although step 319 to collect an identity list 322 for device 101 is depicted as after receiving message 316 (confirming mutual authentication between SE 102 and authentication server 111), step 319 to collect an identity list 322 could be conducted at other times, such as before Figure 3 or after a configuration user 108a takes mobile handset 108 to the approximate physical location for installation or configuration of device 101.

As depicted in Figure 3, configuration server 112 and mobile handset 108 could then establish a secure session setup 321 between the two nodes. Secure session setup 321 could comprise establishing a secure communications link using protocols such as TLS, datagram transport layer security (DTLS), IPSec, a virtual private network (VPN), a secure shell (SSH), or similar networking, transport, or application layer technologies in order to establish secure communications between configuration server 112 and mobile handset 108. Secure session setup 321 could utilize information received by configuration server 112 in message 317 above, such as existing encryption or session keys or certificates for mobile handset 108 from a session 214 (where authentication server 111 and configuration server 112 in turn have a secure session established of session 301). In exemplary embodiments, secure session setup 321 includes mutual authentication for both sides, and may include the transfer of a certificate from each side to the other side. After secure session setup 321, mobile handset 108 can send configuration server 112 a ciphertext.MH 323, where the plaintext inside ciphertext.MH 323 can include the identity list 320 collected by mobile handset in a step 319 above and an exemplary identity list 320 is

depicted in Figure 3. In this manner, a plurality of identifying information regarding device 101 and its operating environment can be securely transferred to configuration server 112.

Configuration server 112 could conduct a step 324 to decrypt and record the identity list 320 received in ciphertext.MH 323. In exemplary embodiments, configuration server 112 records the data from identity list 320 for device 101 in a configuration database 112b as depicted in Figure 1a. The identity list 320 recorded in a step 324 can be utilized in subsequent steps in order to configure device 101 to use transducers 105 with monitored unit 106. Step 324 may also comprise configuration system 114 notifying device owner 122 that SE 102 has been mutually authenticated and identity list 320 has been received. Step 324 could also comprise configuration system 114 (i) processing networks available list 322 in order to select a preferred access network 126 for device 101, and (ii) querying the selected access network 126 for network access credentials 126a for use with the selected access network 126.

Figure 4a

Figure 4a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a mobile handset and a device, in accordance with exemplary embodiments. Before initiating steps and message flows depicted in Figure 4a, mobile handset 108, device 101, and the other elements depicted for system 300 in Figure 4a may previously complete exemplary message flows and steps depicted in Figure 3 above. System 300 can include a configuration server 112, mobile handset 108, and device 101. An exemplary difference between system 300 depicted in Figure 3 and system 300 depicted in Figure 4a can be the authentication steps with authentication server 111 have been completed and thus no additional message flows with authentication server 111 are depicted in Figure 4a. In other words, authentication server 111 from system 300 in Figure 3 may also be in system 300 in Figure 4a, but without any additional messages flowing to or from authentication server 111. Mobile handset 108 can communicate with configuration server 112 via IP network 128, using secure session 321 from Figure 3 above. Mobile handset 108 and SE 102 in device 101 can communicate using NFC session 303, which was also depicted and described above in connection with Figure 3 above.

At step 401, SE 102 can record authentication server 111 has successfully completed authentication above in a step 221d, in addition to a step 221c where SE 102 has successfully completed an authentication with authentication server 111. A step 401 could also include SE 102 authenticating mobile handset 108 before proceeding to subsequent steps depicted in Figure 4a and Figure 4b. SE 102 could request that mobile handset 108

successfully sign a random number such as random1.SE 308 for a signature creation step 220, where SE 102 could verify the signature of random1.SE 308 (or another random number from SE 102) using a signature verification step 221. For this step 401 where mobile handset 108 is authenticated with SE 102, then SE 102 could request a certificate for mobile handset 108, where the certificate authority for the certificate of mobile handset 108 can be verified using an existing certificate stored in SE 102, such as cert.CA1 115 or cert.CA.root 109. A step 401 could also comprise SE 102 utilizing Domain Name System Security Extensions (DNSEC) in order to verify domain names associated with configuration system 114. SE 102 could send DNS requests with associated security extensions to mobile handset 108, which could forward the requests to access network 126 to resolve the DNS requests and provide associated security data such as certificates for DNS servers.

At step 401, SE 102 can also wait for additional signatures of random1.SE 308 from other servers besides authentication server 111 in order to authenticate the other servers, such waiting for a signature of random1.SE 308 or random2.SE (with message 316) from a configuration server 112. In addition, SE 102 can record all used random numbers in nonvolatile memory such as SE flash memory 102d or storage memory 102y, in order to verify that random numbers are not reused and thus increase resistance to replay attacks. A device 101 with SE 102 could also store or record a second set of parameters 407', potentially different than a first set of parameters 407, where device 101 could use the second set of parameters 407's to derive PKI key pairs below such as in a step 410.

At step 402, configuration server 112 can query a configuration database 112a for a set of parameters 407 supported by SE 102. In a system 100 and 300 which supports thousands or more devices 101 and also potentially dozens or more different models manufactures 101x for device 101, then potentially different sets of parameters 407 could be utilized or compatible for different devices 101. A first set of parameters 407 could include or be associated with a second set of parameters 407', and parameters 407' can be used with subsequent keys derived or selected by device 101 for future communications after completion of the steps depicted in Figure 4a. For example, parameters 407' could be used by device 101 with an access network 126 or reporting server 116 as depicted in Figure 5 below. An exemplary "large" device 101 such for an automobile may support PKI key parameters for RSA algorithms with associated longer keys and greater processor requirements (such as exemplary PKI keys of length 2048 or 4096 bits), while an exemplary "small" device 101 controlling a lock on a door could support PKI key parameters for ECC algorithms with lower processor requirements (such as an exemplary PKI key length of 256 bits).

Configuration server 112 could utilize ID.device 101b or ID.SE 102m in order to query any of (i) configuration database 112a, (ii) device manufacturer 101x, or (iii) device owner 122 in order to obtain a set of parameters 407 compatible for device 101 and SE 102. ID.device 101b could be obtained from a message 211 depicted in Figure 2a. At step 5 402, configuration server 112 could also utilize ID.device 101b or ID.SE 102m in order to query any of (i) configuration database 112a, (ii) device manufacturer 101x, or (iii) device owner 122 for PK0.SE 102n and PK0.SE 102n'. Note that configuration system 114 using authentication server 111 could receive PK0.SE 102n and PK0.SE 102n' in conjunction with a step 219a in Figure 2a, where authentication server 111 could query another server or node in a system 100 or a database with data received from the tag 101s. 10

In some exemplary embodiments, configuration server 112 reads a set of parameters 102p from a certificate cert0.SE 102k and selects parameters 407 that are compatible or identical to parameters 102b. In other words, if certificate cert0.SE 102k has a public key PK0.SE 102n with an exemplary ECC public key based on a secp256r1 curve with length 256 bits, then in a step 402 configuration server 112 can determine that a set of 15 parameters 407 should specify an ECC key pair for the secp256r1 curve with length 256 bits. Or a step 402 could comprise configuration server selecting a subset of authentication parameters 111d from Figure 2a. Preferred exemplary embodiments in the present invention support the use of parameters 407 that are different from parameters 111d or 20 parameters 102p, since (i) parameters 102p may be selected by a device manufacturer 101x, but (ii) device owner 122 may desire a different set of parameters 407 be used for device 101 after a configuration step 103 (e.g. different ECC curve names or key lengths or certificate expiration dates or different certificate authorities to sign public keys, etc.)

Note that configuration server 112 could select a first set of parameters 407 for use 25 with PK0.SE 102n and/or PK0.SE 102n', and a second set of parameters 407' (such as with different curves or key lengths) could be used with a second public key for device 101 such as PK1.SE 410a. Parameters 407' could also specify cryptographic algorithms that are different than classical ECC algorithms, such as algorithms for lattice-based cryptography, code-based cryptography, or supersingular isogeny based cryptography, and other 30 possibilities exist as well. Example possible algorithms for parameters 407' include those described in "Post-Quantum Cryptography Standardization" from Wikipedia dated March 12, 2019, which is herein incorporated by reference.

Continuing at step 402, configuration server 112 can derive an ephemeral public key PKe.CS 402a and an associated secret key SKe.CS 402b, using a random number, a 35 key pair generation function, and the set of parameters 407, where obtaining the set of parameters 407 were discussed in the paragraph above. In addition, although PKe.CS 402a

and SKe.CS 402b are described above as “ephemeral” keys, the keys could comprise long-term keys. Further, PKe.CS 402a and SKe.CS 402b could be derived by another server associated with a configuration system 114 and transferred to configuration server 112 before a step 402. Configuration server 112 or configuration system 114 can utilize a
5 random number generator similar to hardware random number generator 102e in Figure 1b in order to derive the random number and input the random number into the key pair generation function. A key pair generation function could be included in application software running on configuration server 112 or an associated server of configuration system 114, and exemplary software for key pair generation algorithm can comprise
10 OpenSSL, crypto++, or Mozilla libraries from Network Security Services.

Other possibilities exist as well for libraries or software for elements of a system 100 or system 300 to derive a PKI key pair. Parameters 407 may specify algorithms and values associated with the PKI key pairs and their use, such as the exemplary parameters 407 depicted in Figure 4d below. Parameters 407 and 407' can specify elliptic curve
15 names (e.g. NIST P-256, sect283k1, sect283r1, sect409k1, sect409r1, etc.), key lengths, a duration for key validity, uses of the PKI key pair such as for key derivation, signatures, encoding rules, message digest algorithms, etc. Parameters 407 and 407' may also include identifying information associate with the PKI key pair such as a sequence number, a serial number, a domain name of the server or element deriving or using the PKI key pair. Other
20 possibilities exist as well for data in a set of parameters 407 without departing from the scope of the present invention.

At step 220e in Figure 4a, configuration server 112 can use a signature creation 220 algorithm as depicted in Figure 2b above in order to create a signature signature1.CS(random1.SE 308, PKe.CS 402a) 404. Note that a signature 404 could also
25 be over parameters 407 and/or parameters 407' as well. In other words, configuration server 112 can sign both (i) the random challenge from SE 102, where authentication server 111 also previously signed the same random challenge, and (ii) the public key PKe.CS 402a generated above in step 402. The signature1.CS 404 can utilize a long-term private key for CS 112 corresponding to a public key in a certificate cert.CS 112c. Although not
30 depicted in Figure 4a, for embodiments where SE 102 sends a second random number random2.SE with the “OK” message in message 316, signature1.CS 404 could be over random2.SE instead of over random1.SE 308.

For embodiments where authentication server 111 and configuration server 112 are combined into the same server or “cloud”-based service or network such as a configuration
35 network 113 in Figure 1a, then step 220e with the associated signature1.CS 404 could be omitted. A step 220e in Figure 4a could comprise configuration server 112 also including

parameters 407 and/or parameters 407' in the "message to sign" for signature1.CS 404. For embodiments where authentication server 111 and configuration server 112 are combined, the IP address utilized by configuration system 114 for message 312 above and message 403 herein could be the same IP address. The use of separate servers for
5 configuration server 112 and authentication server 111 can increase security and scalability for large networks with tens of thousands or more devices 101 for some embodiments.

Configuration server 112 can then send mobile handset 108 a message 403, where message 403 can include or comprise the signature 1.CS (Random1.SE, PKe.CS) 404, a certificate cert.CS 112c, a certificate authority certificate cert.CA1 115, PKe.CS 402a,
10 parameters 407, parameters 407' and ID.Transaction1 405. Parameters 407' in message 403 could be omitted if device 101 records parameters 407' before a message 403, such as described above with a step 401. In other words, either a configuration system 114 or a device 101 can specify the parameters 407' which could be used in a subsequent step 410 with a SK1.SE and PK1.SE. Although not depicted in Figure 4a, message 403 could also
15 include additional parent certificates for cert.CA1 115, such that SE 102 could verify the certificates with a certificate stored in SE 102 such as a root certificate 109.

In exemplary embodiments, configuration database 112a includes the certificates recorded by SE 102 such that configuration server 112 can determine if any parent certificates for cert.CA1 115 are required in order to be verified up through a root
20 certificate recorded in SE 102. In exemplary embodiments, configuration system 114 can (A) query device manufacturer 101x or device owner 122 in system 100 using ID.SE 102m or ID.device 101b for a list of certificates recorded by SE 102 in order to (B) include certificates in message 403 where the public key for a certificate authority matches the root public key for a root certificate 109 installed and recorded by SE 102. ID.Transaction1 405
25 can comprise a preferably unique transaction identity or session identity, such that configuration server 112, mobile handset 108, and SE 102 can track the flows of data and messages between the elements.

At step 408, mobile handset 108 can receive the message 403 using configuration application 108b. Note that message 403 may be transmitted between mobile handset 108
30 and configuration server 112 using secure session 321 and the data in message 403 may be transmitted as a ciphertext. Mobile handset 108 can optionally verify the signature signature1.CS 404 using a signature verification 221 depicted in Figure 2b, although verification by mobile handset 108 may not be required because mobile handset 108 can utilize secure session 321 for communication with configuration server 112. After step
35 408, mobile handset 108 can transmit message 403 with the data described in the paragraph above to SE 102 using NFC session 303, as depicted in Figure 4a. In exemplary

embodiments, the data within message 403 (such as the data depicted in message 403 from configuration server 112 to mobile handset 102) can be sent over the NFC session 303 link, where the message 403 over the NFC session 303 can be formatted according to the data-link layer of NFC session 303. Further a subset or superset of the data for message 403 received by mobile handset 108 could be in message 403 transmitted by mobile handset 108.

At step 409a SE 102 can receive message 403 and record values from message 403 into both volatile memory RAM 102b and storage 102d. Exemplary data recorded by SE 102 in step 409a includes recording in memory by SE 102 both certificates cert.CS 112 and cert.CA1 115. SE 102 can verify certificate cert.CS 112c using certificate cert.CA1 115, and then either (i) verifying cert.CA1 115 with a certificate already recorded in SE 102 before step 409a, such as cert.CA.root 109 (assuming CA1 has its certificate signed by the root certificate authority shown in Figure 1a), or (ii) requesting mobile handset 108 to fetch a parent certificate for cert.CA1 115 that would link cert.CA1 115 to a certificate already recorded and authenticated within SE 102. SE 102 could use a signature verification 221 step in step 409a to verify certificates. In exemplary embodiments, SE 102 records a plurality of “previously approved” root certificates 109 (in Figure 1b) from different certificate authorities, and step 409a could comprise SE 102 selecting a parent root certificate that matches cert.CA1 115. In exemplary embodiments, the plurality of “previously approved” root certificates 109 for SE 102 could be recorded by a device manufacturer 101x during manufacturing of device 101 or before distribution of device 101 to device user 101a. In exemplary embodiments, SE 102 can also query mobile handset 108 for authenticated certificate revocation data for any certificate received in a message 403. SE 102 can proceed with subsequent steps after SE 102 can confirm that certificates received in a message 403 have not been revoked.

In exemplary embodiments, if SE 102 does not record a parent root certificate 109 (or grandparent, or great-grandparent, etc.) for cert.CA1 115 or cert.CS 112, then in step 409a SE 102 can (i) send an error message to mobile handset 108 and (ii) optionally stop the configuration step 103. The error condition to mobile handset 108 could notify configuration user 108a that procedures may be required in order to load a new or different root certificate 109 or parent certificate in order to proceed. The root certificate 109 can be the foundation of security for SE 102. If hackers or adversaries could load a fake root certificate 109, or have SE 102 accept a certificate authority certificate that is not chained to an authenticated root certificate previously installed in SE 102, then potentially any non “genuinely” authenticated public key could be subsequently loaded, and SE 102 could then be considered no longer secure.

At step 409b, SE 102 can process parameters 407 and parameters 407'. SE 102 can verify that parameters 407 and 407' are compatible with the set of cryptographic algorithms 141. For example, if parameters 407 or 407' specify the use of an elliptic curve that is not supported by SE 102, then SE 102 could send mobile handset 108 and error of the unsupported parameters. In step 409b, SE 102 could also select a subset of parameters 407 that are supported by cryptographic algorithms 141 and send mobile handset 108 and configuration server 112 a report of the selected subset. In exemplary embodiments, where parameters 407 and 407' are compatible with cryptographic algorithms 141 and/or certificate parameters 102p then SE 102 accepts parameters 407 and 407' and records them in nonvolatile memory.

At step 221e, SE 102 can verify the signature1.CS 404 using a step 221e as depicted in Figure 2b. SE 102 could input (i) the value for random1.SE 308 issued by SE 102 above message 307 from Figure 3 into the "message to verify" in a signature verification 221 step and also input (ii) the public key ephemeral PKe.CS 402a received in message 403 into in order to verify signature1.CS 404. In this manner, SE 102 can confirm that PKe.CS 402a is authentically from CS 112. Step 221e in Figure 4a can comprise SE 102 using a public key for configuration server 112 from cert.CS 112c. As mentioned above with message 316, if SE 102 issues a second, different random2.SE with message 316 or related messages, then signature1.CS 404 could be over the different random2.SE instead of random1.SE 308 as depicted in Figure 4a. Although not depicted in Figure 4a, in exemplary embodiments signature1.CS 404 verified in a step 221e can also be over parameters 407 and/or 407', and thus SE 102 can know that parameters 407 and/or 407' are authenticated (since they are signed along with random1.SE 308).

At step 410, SE 102 can derive a public key PK1.SE 410a and an associated secret key SK1.SE 410b, using a random number, a key pair generation function, and data from the set of parameters 407'. The parameters 407' could be received in message 403, or also previously stored by device 101 and SE 102 before the receipt of a message 403, as noted above. For some embodiments, PK1.SE 410a could use parameters 407 instead of parameters 407', and the selection of parameters 407 versus 407' could be based on the receipt of parameters 407' in a message 403 or recording of parameters 407' by SE 102 before receipt of a message 403. SE 102 can utilize a random number generator 102e in Figure 1b in order to derive the random number and input the random number into the key pair generation function. A key pair generation function could (i) be included in cryptographic algorithms 141, and (ii) utilize data from parameters 407. SE 102 can then record both PK1.SE 410a and SK1.SE 410b in a nonvolatile memory, such as memory 102d. In exemplary embodiments, SK1.SE 410b is not transferred outside SE 102

(including SK1.SE 410b not being transferred to device 101 outside SE 102) and thus SK1.SE 410b can remain reasonably secure.

SE 102 can conduct a key exchange 411a step with configuration server 112 using PKe.CS 402a and SK0.SE 102y. Details for key exchange 411a are depicted and described in connection with Figure 4b below. As depicted in Figure 4b below, the output of a key exchange 411a step can be a mutually derived symmetric encryption key 442a and MAC key 442b. SE 102 can then perform a signature creation step 220g in order to create signature2.SE 416, where the “message to sign” data in a step 220g includes PK1.SE 410a, and the private key utilized by SE 102 to create signature2.SE can be SK0.SE 102y’. In this manner, (i) the derived public key for SE 102 can be signed by the previously recorded secret key (SK0.SE 102y’), and (ii) configuration server 112 can have access to corresponding PK0.SE 102n’ in order to verify signature2.SE 416. In preferred exemplary embodiments, the parameters 111d used with a signature creation 220g step are equal or equivalent to parameters 102p recorded in cert0.SE 102k’. Note that PK1.SE 410a that is signed in a step 220g may be associated with a different set of parameters 407 than the parameters 102p, such as PK1.SE 410a using the second set of parameters 407’. For a signature creation step 220g, SE 102 could also include a random number received from configuration system 114, such as random1.AS 223, such that signature2.SE 416 can be over at least PK1.SE 410a and random1.AS 223, thereby ensuring a signature 416 is not replayed.

SE 102 can then create a ciphertext1.SE 415 using an encryption step 412a, where encryption step 412a is depicted and described in connection with Figure 4b below. SE 102 can use the derived symmetric encryption key 442a and MAC key 442b (from key exchange 411a in Figure 4b below) with encryption step 412a. The encrypted data in ciphertext1.SE 415 can include the derived public key PK1.SE 410a and also optionally configuration data for SE 102 such as data config0.SE 102aa. Data for config0.SE 102aa was described above in connection with Figure 1b, and in exemplary embodiments, config0.SE 102aa includes a list of all root certificates recorded by SE 102. Including both (i) ciphertext1.SE 415 of PK1.SE 410a and (ii) signature2.SE 416 may appear redundant to those of ordinary skill in the art, since a subsequent decryption of ciphertext1.SE 415 by configuration server 112 would normally mean that only SE 102 sent the ciphertext1.SE 415 (encrypted using SK0.SE 102y). But, creating a separate signature2.SE 416 for PK1.SE 410a can allow configuration server 112 to send the signature2.SE 416 to other servers, such a certificate authority 115 as depicted in Figure 4a. Additional details for using signature2.SE 416 are described below in connection with Figure 5.

SE 102 can then send mobile handset 108 a message 413, where message 413 can include ID.Transaction1 405, ciphertext1.SE 415, and signature2.SE 416. As depicted in Figure 4a, ciphertext1.SE 415 comprise plaintext values or data for PK1.SE 410a and config0.SE 102aa, where the plaintext values are encrypted with symmetric encryption key 442a. Signature2.SE 416 can be over derived public key PK1.SE 410a. ID.transaction1 405 can correspond to the transaction ID received in a message 403 and could comprise a session ID or token for SE 102 and configuration server 112 to track the data flow. Although not depicted in Figure 4a, signature2.SE 416 can also be over the transaction ID 405 in addition to PK1.SE 410a. As depicted in Figure 4a, message 413 can be sent over NFC session 303 to mobile handset 108. Mobile handset 108 can then forward the data from message 413 to configuration server 112 using secure session 321. For embodiments where device 101 and/or SE 102 stores parameters 407' before the receipt of a message 403 and uses the parameters 407' to generate SK1.SE 410b and PK1.SE 410b, then message 413 can also include the parameters 407'.

Before or after receipt of message 413, configuration server can conduct a key exchange 411b step, which is depicted and described in connection with Figure 4b below. A key exchange 411b step can allow configuration server 112 to mutually derive a same symmetric encryption key 442a that was derived by SE 102 in a key exchange step 411a. For step 411b, configuration server 112 can use SKe.CS 402b and PK0.SE 102n in order to derive symmetric encryption key 442a. Figure 4a depicted step 411a before receipt of message 413 but step 411a could be conducted after receipt of message 411a. After receipt of message 413, configuration server 112 can then perform a decryption 412b step of received ciphertext1.SE 415 using the derived symmetric encryption key 442a, and the operation of a decryption 412b step is depicted and described in connection with Figure 4b below. After step 412b, configuration server 112 can read the plaintext values of PK1.SE 410a and config0.SE 102aa. Configuration server 112 can then also optionally perform a signature verification step 221g, where signature2.SE 416 over at least PK1.SE 402a can be verified. Note that configuration server 112 can optionally omit signature verification step 221g since successful decryption of ciphertext1.SE 415 could only normally be conducted with an authentic SE 102 (e.g. the entity holding SK0.SE 102y). However, other elements in a system 100 and other systems herein can later use signature2.SE 416 to authenticate that PK1.SE 410a is truly from SE 102.

At step 417, configuration server 112 can send a certificate authority 115 for configuration system 114 a message with ID.SE 102m, PK1.SE 410a and signature2.SE (PK1.SE 410a) 416. Although not depicted in Figure 4a, the message to certificate authority 115 could be over a secure session similar to secure session 321. Although not

depicted in Figure 4a, the message to certificate authority 115 could also include (i) data for (a) parameters 407 or 407' for use with PK1.SE 410a and (b) signature2.SE 416, and (ii) a certificate cert0.SE 102k or cert0.SE 102k'. Certificate authority 115 could then conduct a signature verification step 221g for signature2.SE 416 using PK0.SE 102n from
5 cert0.SE 102k (or a PK0.SE 102n' from a cert0.SE 102k' for digital signatures), such that PK1.SE 410a is authenticated for being from SE 102 using ID.SE 102m (since ID.SE 102m is preferably in certificate cert0.SE 102k'). Once signature2.SE 416 over PK1.SE 410a is verified, certificate authority 115 can conduct a step 418 to create certificate cert1.SE 419, where cert1.SE 419 can be signed by certificate authority 115. Note that for embodiments
10 where signature 416 is also over a random number from configuration system 114, such as random1.AS 223, then (i) the message in a step 417 can include the random number, such that CA 115 can verify signature2.SE 416 is over both PK1.SE 410a and the random number from configuration system 114.

Step 418 can use parameters 407' in order to create certificate cert1.SE 419, and in
15 exemplary embodiments parameters 407' are different than parameter 102q, including specifying (i) different key lengths in bits, (ii) different named ECC curves, and (iii) different validity dates. An exemplary certificate cert1.SE 419 for SE 102 using derived PK1.SE 410a is depicted in Figure 4d below. Certificate authority 115 can send cert1.SE 419 to configuration server 112. Cert1.SE 419 for SE 102 can facilitate SE 102 to begin
20 using PK1.SE 410a and SK1.SE 410b in an authenticated manner with other elements of a system 100, including reporting system 120 or access network 126. The use of cert1.SE 419 can provide additional security for a device owner 122 or device user 101a, since (i) SK1.SE 410 has been derived after receipt of device 101 from device manufacturer 101x, and also with (ii) the security for device 101 no longer depending on SK0.SE 102y and
25 SK0.SE 102y', which have physically been in control of other parties besides device owner 122 or device user 101a before delivery of device 101.

At step 420, configuration server 112 can receive cert1.SE 419 and conduct a signature verification step 221 over the signature created by step 220 depicted in Figure 4d. Configuration server 112 can also query configuration database 112a for parent certificates
30 for cert1.SE 419 and cert.CA1 115 that would link to a root certificate recorded in SE 102. In other words, configuration server 112 in a step 420 finds and records intermediate certificate authority certificates that could be verified by a root certificate listed in config0.SE 102aa. As mentioned above, config0.SE 102aa received by configuration server 112 can contain a list of all root certificates recorded by SE 102. Further, at step
35 420, configuration server 112 can query for intermediate certificates that link cert.CA2 121 to a root certificate listed in config0.SE 102aa, such that SE 102 can verify a certificate

from reporting system 120 with a root certificate already installed in SE 102, such as at least one root certificate 109 burned into EEPROM 102j during manufacturing.

For embodiments where PK1.SE 401a use a different set of cryptographic parameters 407' than parameters 407, then configuration server 112 can conduct a second key derivation step 402' with the parameters 407' in order to generate PKe.CS 402a' and SKe.CS 402b'. At a second key exchange 421a step, configuration server 112 could conduct second a key exchange where (i) the derived, authenticated, and previously signed PK1.SE 410a is used for the key exchange instead of the original PK0.SE 102n, and (ii) the SKe.CS 402b' from a step 402'. Details for the second key exchange 421a step are depicted and described in connection with Figure 4c below. For embodiments where PK1.SE 410a use parameters 407, then configuration server 112 could continue to utilize the ephemeral SKe.CS 402b in a key exchange step 421a. The output of a key exchange step 421a can be a new, second symmetric encryption key 442b and also a new MAC key 442b. In other words, the second key exchange step 421a can be utilized so that configuration server 112 can derive the new symmetric encryption key 442b using PK1.SE 410a and SKe.CS 402b'.

The present invention as depicted in Figure 4a provides an embodiment for security that may also be generally applicable for a server communicating with a node, even outside the use of a configuration 103 step. Specifically, the configuration server has (i) derived an ephemeral key PKI key pair (e.g. PKe.CS 402a and SKe.CS 402b), where the ephemeral key is signed with a static server private key, (ii) conducted a first key exchange 411b with the derived SKe.CS 402b for a first symmetric key 442a, (iii) received a derived PK1.SE 410a from SE 102 where PK1.SE 410a can be either (iii.a) encrypted with the first symmetric key 442a, or (iii.b) verified with PK0.SE 102n', or (iii.c) both iii.a and iii.b, (iv) conducted a second key exchange 421b using the second ephemeral SKe.CS 402b' and the received, derived PK1.SE 410a in order to obtains a second symmetric key 442b, and then (v) encrypting ciphertext with the second symmetric key 442b. In other words, the steps the configuration server 112 herein has taken can be applicable generally to a server communicating with a node and using both static and ephemeral PKI keys for the server to derive a first symmetric encryption key, and then securely receiving a derived public key from the node and using that derived public key with the ephemeral keys to derive a second symmetric encryption key.

At step 422, configuration server 112 with other elements in configuration system 114 can select or obtain network access credentials 126a. Configuration server 112 could query configuration database 112a using the networks available list 322 depicted and described in connection with Figure 3 above, where the networks available list 322 provide

information about wireless networks around device 101. A response to a query to configuration database 112a could provide a selected access network 126 that is preferred based on criteria such as bandwidth cost, expected energy or power requirements (i.e. 3G/4G may use more power than LPWAN), signal strength measurements for networks available list 322, and commercial terms such as agreements between device owner 122 and networks in a networks available list 322. Configuration server 112 could also query other servers in a configuration system 114 or a system 100 in order to obtain a selected access network 126 from networks available list 322 in addition to network access credentials 126a for access network 126.

In an exemplary embodiment, network access credentials 126a received by configuration server 112 or configuration system 114 are encrypted using either PK0.SE 102n or PK1.SE 410a before being received by configuration server 112. A second symmetric encryption key 442 (with a different value than the first symmetric encryption key 442a below in Figure 4b) could be asymmetrically encrypted using PK0.SE 102n or PK1.SE 410a, and the network access credentials 126a could be encrypted with the second symmetric encryption key 442a by an access network 126. In exemplary embodiments, the second symmetric encryption key 442 can be asymmetrically encrypted by a configuration system using PK1.SE 410a and Elgamal asymmetric encryption. Other possibilities exist as well such as another element besides configuration server 112 deriving a second symmetric encryption key 442 using PK0.SE 102n or PK1.SE 410a and then encrypting the network access credentials 126a using the derived second symmetric encryption key 442. In that manner, the encrypted network access credentials 126a can be encrypted uniquely for SE 102 and cannot be read by configuration server 112 or mobile handset 108. In some exemplary embodiments, the network access credentials 126a can include an identity such as an international mobile subscriber identity (IMSI) and a shared secret key "K", and each of the IMSI and the key K can be asymmetrically encrypted with PK1.SE 401a, where device 101 with SE 102 could asymmetrically decrypt the IMSI and key K with SK1.SE 401b.

At step 423, configuration server 112 can collect a software package 104q for device 101 and SE 102. An exemplary software package 104q for device 101 is depicted and described in Figure 5 below. Configuration server 112 could use data from an identity list 320 to query other servers from a system 100 in order to obtain the software. Different components of software package 104q could come from different servers or elements in a system 100. Configuration server 112 can also use data from configuration file config0.SE 102aa in order to select software for software package 104q. The data collected by or queried by configuration server 112 could be stored in a configuration database 112a. An

updated device 101 operating system 101e could be obtained from a device manufacturer 101x. Configuration data config.device 101r could be obtained from servers associated with either device owner 122 or device manufacturer 108x.

At step 423 configuration server 112 can also use ID.transducer from identity list 320 to query a configuration database 112a or configuration database 112b for in order to obtain a configuration file or dataset for device 101 to use with transducer 105, and the dataset could comprise config.TR1 105e. Although not depicted in Figure 1a a system 100 could include a transducer manufacturer similar to device manufacturer 101x, and a configuration server 101 could use ID.transducer to query the transducer manufacturer for data to be included in config.TR1.105e. Data for config.TR 105e could comprise a transducer electronic datasheet (TEDS). At step 423 configuration server 112 can use ID.MU to query a configuration database 112a or configuration database 112b for in order to obtain a configuration file or dataset for device 101 to use with monitored unit 106, and the dataset could comprise config.MU1 106a. The dataset for config.MU1 106a can specify parameters associated with monitored unit 106, such as timer values for device 101 to collect data, a range of maximum or minimum values an actuator for transducer 105 could use with monitored unit 106. In exemplary embodiments, the dataset for config.MU1 contains thresholds for alarm values or conditions, such that if device 101 detects signals outside acceptable levels and within an alarm range in config.MU1, then device 101 immediately reports the alarm condition to reporting server 116 and/or takes other actions as well, including potentially adjusting a transducer output 105c to mitigate the alarm condition. Other possibilities exist as well for data within config.MU 106a without departing from the scope of the present invention.

Further, at step 423 configuration system 112 could request device owner 122 to specify the reporting system 120 for device 101, and subsequently (A) configuration server 112 could send data from identity list 320 or config0.SE 102aa to reporting system 120 and (B) reporting system 120 could send back a dataset to configure device 101 to work with reporting system 120, and the dataset could comprise config.reporting-system 120a. Data within config.reporting-system 120a could specify domain names or IP addresses of reporting server 116, credentials for device 101 to utilize with reporting server 120 such as a user name, password, encryption keys, or a public key, reporting timers, etc. In some exemplary embodiments, network access credentials 126a could be received by configuration server 112 in a step 423 from reporting system 120. In other exemplary embodiments where PKI-based authentication is used for connectivity of device 101 to access network 126, then a step 423 could also comprise configuration server 112 sending access network 126 the certificate cert1.SE 419. Data within config.reporting-system 120a

could also specify cryptographic parameters similar to parameters 407 such as algorithms to utilize, key lengths, curve names, padding schemes, encoding rules, versions of standards to implement (such as TLV v2 versus TLS v3). Configuration server 112 could also query device owner 122 in order to obtain some data for config.reporting-system 120a.

5 In addition, configuration server 112 could query configuration user 108a at mobile handset 108 for additional data or user input if the software package 104q could not be automatically assembled or fully created. Note that software package 104q could include sub-packages, such as a sub-package 104z, where sub-package 104z could depend or be determined based on transducers 105 attached to device 101, or monitored unit 106, or
10 reporting system 120. In an exemplary embodiment, identity list 320 could specify a transducer identity ID.transducer that requires a specific transducer library 105q (or driver) for device 101 with OS 101e to support a selected transducer 105. Configuration server 112 could query a transducer manufacturer for the most current or appropriate transducer library 105q for device 101, and include that transducer library 105q in a software package
15 104q.

In exemplary embodiments a software package 104q can include a set of configuration test vectors 104x, which can comprise software for verifying the functionality of transducers 105 with transducer bus 101i and transducer library 105q. A configuration test vector 104x could provide an instruction to verify signal strength for data
20 received, noise levels for data received, or dynamic range for signals output, etc. Other possibilities exist as well for a configuration server 112 or configuration system 114 to assemble software for device 101 in a software package 104q, without departing from the scope of the present invention. After software package 104q is assembled, configuration server can calculate the size of software package 104q, which could comprise the value
25 package-size.CS 428, where package-size.CS represents the size of the software package 104q in bytes.

At step 424a, configuration server 112 can encrypt data for SE 102 using the derived symmetric encryption key 442b from a previous step 421a. An encryption step 424a is depicted and described in connection with Figure 4c below, and the encryption step
30 424a can create a ciphertext1.CS 427. At step 424a, plaintext input into a symmetric encryption algorithm 444a could include at least one of certificates cert1.SE 419, cert.CA2 121, cert.CA3 123, device 101 configuration data config.TR1 105e, config.MU1 106a, config.device 101r, and config.reporting-system 120a. In other words, plaintext input could include one or more of certificates cert1.SE 419, cert.CA2 121, cert.CA3 123,
35 device 101 configuration data config.TR1 105e, config.MU1 106a, config.device 101r, and config.reporting-system 120a. The data for plaintext in a step 424 can comprise a device

configuration. In addition, plaintext input into a symmetric encryption algorithm 444a at a step 424a could include identity list 320 and network access credentials 126a (or an encrypted network access credentials 126a, where the encrypted network access credentials were encrypted using a public key for SE 102). A step 424a could also include
5 configuration server 112 using plaintext of additional parent certificates for cert.CA1 121 and/or cert CA3 123 that would be signed by an existing cert.CA.root 109 recorded by device 101. The parent certificates would be determined in the prior step 420 as described above.

Configuration server 112 can then send message 425 to mobile handset 108, where
10 message 425 can contain an ID.transaction2 426, PKe.CS 402a', ciphertext1.CS 427, where ciphertext1.CS 427 can include the ciphered data described in the above paragraph and depicted in both Figure 4a and Figure 4c. Other or additional data could be included in a ciphertext1.CS 427 as well, such as the ID.transaction2 426, and identity of configuration server 112 or SE 102 or device 101, or other values as well. Message 427 can also include
15 the value package-size.CS 428, and package-size.CS 428 may also optionally be in ciphertext1.CS 427. In exemplary embodiments, package-size.CS 428 is preferably outside ciphertext1.CS 427 in order for mobile handset 108 to read the value and make decisions or prompt configuration user 108a based on the value of package-size.CS 428. If package-size.CS 428 was only inside ciphertext1.CS 427, then mobile handset 108 would not
20 normally be able to read the plaintext data since mobile handset 108 does not hold or record a symmetric encryption key for ciphertext1.CS 427. Upon receiving message 425, mobile handset 108 can forward data in message 425 to SE 102, as depicted in Figure 4a.

At step 429, mobile handset 108 can process package-size.CS 428 in order to make a decision about subsequent data transfer between mobile handset 108 and device 101
25 through NFC session 303. Although depicted in Figure 4a as occurring after mobile handset 108 sending message 425, a step 429 could occur before the mobile handset 108 sends message 425 to SE 102. Configuration application 108b can read package-size.CS 428 to determine if the upcoming software package size 104a in bytes should be transmitted to SE 102 via either (A) the existing NFC session 303 or alternatively (B) via a
30 new, different Wi-Fi or Bluetooth session between mobile handset 108 and device 101. For example, if package-size.CS 428 is less than an exemplary one megabyte, then a configuration user 108a for mobile handset 108 could reasonably wait for the future software package 104q to be transmitted over NFC session 303 (such as less than an exemplary 30 seconds). However, if package-size.CS 428 is relatively large, such as
35 greater than an exemplary ~15 megabytes, then a configuration user 108a may prefer to setup a Wi-Fi or Bluetooth session with device 101 in order to transfer the upcoming

software package 104q. A step 429 could also comprise a display on mobile handset 108 prompting configuration user 108a for a decision or threshold for determining the use of Wi-Fi or Bluetooth (instead of using NFC session 303) in order to transfer upcoming software package 104q based on package-size.CS 428.

5 In exemplary embodiments, a step 429 can take place before mobile handset 108 sends message 425, and message 425 to SE 102 can include a signal or command regarding the use of Wi-Fi or Bluetooth to download software package 104q. Note that SE 102 could separately make a decision regarding the use of Wi-Fi or Bluetooth to download software package 104q based on package-size.CS 428, where SE 102 can consider factors that may
10 not be available to configuration application 108b, such as the Wi-Fi and Bluetooth capabilities of device 101, internal security policies of device 101 and SE 102, etc. Figure 4a depicts an embodiment where SE 102 makes the decision regarding physical layer to utilize for transfer of upcoming software package 104q using a step 431 below. However, the present invention also contemplates that mobile handset 108 can make the decision
15 regarding physical layer to utilize for upcoming transfer of software package 104q as well for other embodiments. Further, configuration server 112 could make the decision based on parameters in a configuration database 112a, such as a maximum time allowed for transfer of software package 104q through NFC session 303, such as an exemplary maximum time of 120 seconds before an alternative physical layer to NFC would be
20 attempted such as Wi-Fi or Bluetooth.

After receipt of message 425, at step 421b, SE 102 can derive symmetric encryption key 442b as depicted and described in connection with Figure 4c below. Note that SE 102 utilizes PKe.CS 402a' and SK1.SE 410b as input into the key exchange algorithm 447. In other words, SE 102 can utilize the secret key SK1.SE 410b derived in
25 step 410 with PKe.CS 402a' received in message 425 above in order to obtain symmetric encryption key 442b. In this manner, SE 102 and device 101 can optionally end dependence on a SK0.SE 102y that likely had been physically been outside the control of device owner 122 and device user 101a. In exemplary embodiments, although SK1.SE 410b provides an updated secret key, SE 102 continues to record SK0.SE 102y with
30 corresponding cert0.SE 102k, such that use of SK0.SE 102y could be enabled for a "rollback" or "fallback" scenario such as if nonvolatile memory 102d storing SK1.SE 410b gets corrupted or erased. For embodiments where parameters 407 are used with SK1.SE 401b, then a step 421b can be conducted with PKe.CS 402a from a message 403 and the use of a PKe.CS 402a' from a message 425 could be omitted.

35 SE 102 can then use a decryption 424b step with the derived symmetric encryption key 442b in order to decrypt the ciphertext1.CS 427 and read the plaintext inside

ciphertext1.CS 427. Although not depicted in Figure 4a but depicted in Figure 4c, a message 425 with ciphertext1.CS 427 can also include an initialization vector 445b. As contemplated for all messages depicted in the present invention, a message with a ciphertext can also include in initialization vector 445 and a message authentication code 446. At step 430 SE 102 can record the decrypted plaintext data from message 425 in nonvolatile memory 102d, including the new cert1.SE 419, which can comprise the new, primary certificate for SE 102 and cert0.SE 102k can be considered deprecated at step 430. In exemplary embodiments the new cert1.SE 419 received by SE 102 can utilize a different set of parameters 407 than cert0.SE 102k. For example, if (a) a device manufacturer 101x may use a different ECC named curve than configuration system 112, then (b.1) a new cert1.SE 419 would normally be required for file transfers with configuration system 112, and (b.2) different values for parameters 407 can be utilized in cert1.SE 419 than cert0.SE 102k. The selection of a set of parameters 407 by a configuration server 112 were depicted and described in connection with step 402 above.

Also at step 430, SE 102 can verify certificates received in message 425, as well as check a certificate revocation list for both cert1.SE 102 and any parent certificates. In exemplary embodiments, (X) all parent certificates through a root certificate are included in message 425, for all certificates received such as cert1.SE 419, cert.CA2 121, cert.CA3 123 in order for (Y) the public key in a root certificate 109 recorded in device 101 or SE 102 before configuration step 103 can be used to verify all the certificates received in message 427. Further, in exemplary embodiments contemplated herein, each time a node or component is described as checking verifying a certificate, the node can also check with a certificate revocation list (CRL) recorded in the certificate to verify that the certificate remains valid before an expiration date. A step 430 can also comprise SE 102 sending a request through NFC session 303 to configuration server 112 to check CRL of certificates received in a message 425. Or, in other exemplary embodiments, a message 425 can include a confirmation that CRL for certificates in message 425 have been checked, and in step 430 SE 102 can also verify that message 425 includes the confirmation that CRL for certificates have been checked by configuration system 114. In another embodiment, any certificate received in a ciphertext1.CS 427 is accepted by SE 102 since SE 102 has authenticated configuration server 112 using random1.SE 308 above in Figure 3.

In exemplary embodiments for a step 430 depicted in Figure 4a, (A) message 425 includes a certificate received by SE 102 that cannot be verified by SE 102 using an existing cert.CA0.root 109 inside SE 102 (including possibly due to unsupported parameters 407), but (B) SE 102 accepts the certificate since it was received from an authenticated server (e.g. configuration server 112). In another exemplary embodiment, SE

102 accepts a second, new root certificate cert.CA1.root and/or cert.CA2.root in a ciphertext1.CS 427 since the message 425 and ciphertext is from configuration server 112, which has been authenticated. The inclusion of a second, new root certificate cert.CA1.root and/or cert.CA2.root in a message 425 can be secure for device 101 and SE 102 since the certificates are within a ciphertext1.CS 427 and thus cannot feasibly be altered or tampered with by mobile handset 108 or other nodes. Note that Figure 4a and related figures depict mobile handset transmitting ciphertext across a local wireless link such as NFC session 303.

Continuing at step 430, SE 102 can continue processing plaintext data from message 425 while loading the data into memory and also verifying the data. In exemplary embodiments, SE 102 applies network access credentials 126a received in message 427, as well as configuration parameters received, such as config.device 101r. Note that device 101 with radio 101z may utilize network access credentials 126a, and for those embodiments then SE 102 can pass network access credentials 126a and/or config.device 101r to device 101 via bus controller 102h and device data bus 101g. Device 101 could store network access credentials 126a and config.device 101r in nonvolatile memory storage 101f. In exemplary embodiments, step 430 also comprises device 101 rebooting in order to apply config.device 101r and network access credentials 126a. In this embodiment, NFC session 303 may temporarily close but reopen upon successful reboot of device 101 and SE 102.

In exemplary embodiments, network access credentials 126a received in a message 425 and includes at least one additional, backup set of network access credentials 126a, such that device 101 can connect with a second access network 126 if the first access network 126 becomes unavailable. The second or backup set of network access credentials 126a could be (i) from configuration server 112 and (ii) selected by configuration server 112 using the networks available list 322 in Figure 3. Step 430 may also comprise device 101 using config.reporting-system 120a in order to communicate with reporting system 120 while activating and accessing an access network 126 using the received network access credentials 126a. In this manner, device 101 can perform an “end-to-end” test to verify the supplied configuration values in a message 425 will provide network connectivity to enable the flow of transducer data 125 from device 101 to reporting server 116 as depicted in Figure 1a. For embodiments where device 101 includes a WAN radio such as a 4G, 5G, 6G radio, etc., then network access credentials 126a could be used by device 101 to connect with an access network 126. Device 101 could receive subsequent configuration data such as that depicted in Figure 5 from access network 126.

At step 431, SE 102 can use package-size.CS 428 to determine if NFC session 303 or a separate physical layer should be used for downloading the upcoming software package 104q. An exemplary decision process for SE 102 using package-size.CS 428 in a step 431 was also described above in connection with step 429. Separate wireless physical
5 layers that SE 102 or device 101 could utilize for downloading software package 104q could be any of Wi-Fi, Bluetooth, a land mobile network such as 4G, 5G, etc., or a LPWAN, and other possibilities exist as well without departing from the scope of the invention. The selection of a physical layer by SE 102 for transfer of a software package 104q based on package-size.CS 428 could comprise a networking command 431a. The
10 exemplary embodiment depicted in Figure 4a illustrates the use of Wi-Fi with mobile handset 108 for download of software package 104q and an associated networking command 431a for use of Wi-Fi with mobile handset 108, but other possibilities exist as well.

Device 101 with SE 102 could connect with a Wi-Fi network from an access point
15 in the vicinity of installed device 101, such as the exemplary Wi-Fi networks detected and depicted for a networks available list 322 in Figure 3 (where network access credentials 126a could be for the exemplary Wi-Fi network in the list 322). Alternatively, device 101 with SE 102 could use a set of network credentials 126a in message 425 to connect with a land mobile network as access network 126 and subsequently download software package
20 104q via the land mobile network, and in this case the networking command from device 101 and SE 102 could specify use of the land mobile network for software package 104q. Figure 4a depicts a decision in a step 431 where SE 102 selects activation of a Wi-Fi radio in device 101, such as radio 101z, in order to download software package 104q. As mentioned above, any of configuration server 112, mobile handset 108, configuration user
25 108a, or SE 102 could make the selection for the physical layer to utilize for software package 104q, or they could make the selection in conjunction.

At step 432, SE 102 can generate a configuration report 432a for configuration system 112 regarding the installation and application of certificates, configuration files, and network parameters and credentials received in message 425. For example, report 432a
30 could confirm the success, bandwidth, signal strength, and other values from using both a primary and a backup set of network access credentials 126a with selected access networks 126 from the networks available list 322. Further report 432a can confirm the success, errors, or other states measured from device 101 implementing the configuration parameters for reporting system 120 in the data config.reporting-system 120a, such as the
35 ability to send or receive transducer data 125. Config.reporting-system 120a can include credentials for device 101 to connect with reporting system 120, such a certificate for

reporting server 116. Report 432a could also include data regarding success or errors related to loading and verifying the exemplary certificates in ciphertext1.CS 427 of cert1.SE 419, cert.CA2 121, cert.CA3 123, etc. Report 432a can also include available memory for SE 102 or device 101 (e.g. memory 102b, 102d, 102x, or 101f), such that
5 configuration server 112 can confirm resources are sufficient for the planned transfer of software package 104q. Other possibilities exist as well for a report 432a to contain data pertaining to the status and resources of device 101 or SE 102 without departing from the scope of the present invention.

At step 424c, SE 102 can use an encryption step equivalent to encryption step 424a
10 depicted in Figure 4c in order to create a ciphertext2.SE 434. The symmetric encryption key 442b from a prior step can continue to be utilized for encryption step 424c. The plaintext inside ciphertext2.SE 434 can include a networking command 431a which was determined in step 431 above, and the depicted embodiment shows a network command 431a of "Wi-Fi start", or for mobile handset 108 to activate a Wi-Fi connection for device
15 101 in order for SE 102 to receive software package 104q. The plaintext inside ciphertext2.SE 434 can also include a report 432a, where an exemplary report 432a was described in the above paragraph. SE 102 can then send mobile handset 108 a message 433, where message 433 can contain ID.transaction2 426 and ciphertext2.SE 434. Although the exemplary embodiment depicted in Figure 4a has SE 102 sending network
20 command 431a in an encrypted format that cannot normally be read by mobile handset 108, the network command 431a could also be sent at plaintext to mobile handset 108.

Mobile handset 108 can forward message 433 to configuration server 112 and then also conduct a step 435. In step 435, mobile handset 108 can store or backup the currently applied network access credentials for Wi-Fi clients when mobile handset 108 operates as a
25 Wi-Fi access point or "hotspot". In other words, a configuration user 108a may have mobile handset 108 operate as a Wi-Fi access point with specified credentials outside of the purpose of configuring a device 101 (such as when configuration user 108a has mobile handset 108 at their home or office). At step 435 the existing credentials for mobile handset 108 to operate as that Wi-Fi access point are backed up, and step 425 enables (i) a
30 new set of access point credentials to be temporarily utilized for a Wi-Fi access point created by mobile handset 108, where (ii) the "backed up" Wi-Fi access point credentials from this step 435 can be later restored for mobile handset 108.

Configuration server 112 can receive message 433 and conduct a series of steps to process the data in message 433. Configuration server 112 can conduct a decryption step
35 424d in order to convert ciphertext2.SE 434 into plaintext and read networking command 431a and report 432a. The decryption step 424d can correspond to decryption step 424b

depicted and described in connection with Figure 4c below, such as using the same symmetric decryption key 442b that was previously derived by configuration server in a step 421a also depicted in Figure 4c below.

At step 436 configuration server 112 can process networking command 431a and
5 report 432a. Data from report 432a can be recorded in a database 112b. Configuration server 112 can send confirmation data or portions of report 432a to other elements in system 100, such as informing device owner 122 or reporting system 120 or access network 126 of the preliminary configuration of device 101, where the preliminary configuration can be a check that configuration files, certificates and network access credentials delivered
10 in a message 425 were properly received and loaded by device 101 and/or SE 102. In an exemplary embodiment, step 436 could include configuration system 114 notifying reporting system 120 that device 101 should be activated (since the completion of a configuration step 103 is pending). Using report 432a, configuration server 112 can also evaluate or confirm that certificates, configuration data, and network access credentials
15 delivered in the previous message 425 were properly received and applied.

In a step 436 configuration server 112 could use networking command 431a and report 432a to determine that mobile handset 108 should activate a Wi-Fi access point for download of software package 104q to SE 102, instead of continuing to use NFC session 303. In a different exemplary embodiment than that depicted in Figure 4a, configuration
20 server 112 could use networking command 431a and report 432a to determine that SE 102 and device 101 should use a land mobile network such as access network 126 in order to download software package 104q. In yet another exemplary embodiment, configuration server 112 could use networking command 431a and report 432a to determine that SE 102 and device 101 should continue using the existing NFC session 303 in order to download
25 the software package 104q.

At step 437 configuration server 112 can select the access point name (e.g. SSID) and password, as well as other parameters such as security schemes (e.g. WPA2 or WPA3, etc.) for mobile handset 108 to utilize in order to have device 101 connect with mobile handset 108 over Wi-Fi. The data for mobile handset to operate a Wi-Fi access point can
30 be recorded in a dataset Wi-Fi-AP.MH 437a. Configuration server 112 can then conduct an encryption step 424e to create ciphertext2.CS 441, where the plaintext comprises dataset Wi-Fi-AP.MH 437a. Configuration server can utilize previously derived symmetric encryption key 442b to convert the plaintext into the ciphertext2.CS 441. Step 424e could correspond to encryption step 424a depicted in Figure 4c below. Configuration server 112
35 can then send message 438 to mobile handset 108, where message 408 includes dataset Wi-Fi-AP.MH 437a. Note that dataset Wi-Fi-AP.MH 437a can be encrypted inside secure

channel 321 between configuration server 112 and mobile handset 108, so another layer of encryption can be omitted for message 438. Mobile handset 108 would not normally be able to read ciphertext communicated between configuration server 112 and SE 102 as depicted in Figure 4a. Mobile handset 108 can record and process the data received in message 438.

Configuration server 112 can then send mobile handset 108 another message 440, where message 440 includes an ID.transaction3 439 and the ciphertext2.CS 441, where the creation of ciphertext2.CS 441 was described in the above paragraph. Mobile handset 108 can receive message 440 and subsequently send the data for message 440 to SE 102 and SE 102 can receive message 440. SE 102 can then conduct a decryption step 424f in order to convert ciphertext2.CS 441 into plaintext and read dataset Wi-Fi-AP.MH 437a. Decryption step 424f can correspond to decryption step 424b depicted and described in connection with Figure 4c below, where SE 102 can utilize the derived symmetric encryption key 442b. SE 102 could process the decrypted plaintext Wi-Fi-AP.MH 437a in order to know the credentials to use with a Wi-Fi radio 101z for device 101 and connect with mobile handset 108 using Wi-Fi. Data from the decrypted plaintext Wi-Fi-AP.MH 437a can be stored in nonvolatile memory for SE 102. The successful processing of Wi-Fi-AP.MH 437a by SE 102 could also be a signal for SE 102 to activate Wi-Fi for device 101, and utilize the credentials within Wi-Fi-AP.MH 437a.

At step 441, SE 102 can instruct device 101 to activate Wi-Fi through a radio 101z in order for the hardware and software for device 101 to (i) enable Wi-Fi session 503 depicted below in Figure 5, and (ii) receive the additional data and software package 104q to complete a configuration step 103 as depicted in Figure 1a. In other words, Wi-Fi-AP.MH 437a can provide information (i) for mobile handset 108 to activate a Wi-Fi access point with the credentials for an access point and (ii) for device 101 to activate a Wi-Fi client with the credentials for an access point. Wi-Fi-AP.MH 437a can also provide information such as a maximum power level to utilize in a Wi-Fi session 503, such as a typically lower power level due to the close proximity of the two devices. In exemplary embodiments, Wi-Fi-AP.MH 437a specifies a power level less than 1 milliwatt for either or both nodes maximum transmit power, although other possibilities exist as well.

In an alternative exemplary embodiment, device 101 can use radio 101z to activate a Wi-Fi access point with the credentials for Wi-Fi-AP.MH 437a, and then mobile handset 108 can activate a Wi-Fi client with the credentials for Wi-Fi-AP.MH 437a. In another embodiment for the present invention, either mobile handset 108 or device 101 can activate a Wi-Fi access point and the other side can operate as the Wi-Fi client. Benefits of device 101 activating the Wi-Fi access point for a Wi-Fi session 503 below in Figure 5 is that

device 101 can potentially retain greater control and thus security for the Wi-Fi session 503, but device 101 activating the Wi-Fi access point is not required in order to gain benefits from the present disclosure. Further, although Wi-Fi-AP.MH 437a is depicted in Figure 4a as credentials and parameters for a Wi-Fi network, other local wireless connectivity technologies could be utilized as well, including Bluetooth. For embodiments where NFC session 303 remains open and utilized for the transfer of software package 104q, then the use of Wi-Fi-AP.MH 437a and activation of a Wi-Fi access point and client between mobile handset 108 and device 101 can be optionally omitted.

10 Figure 4b

Figure 4b is a series of flow charts illustrating exemplary steps for nodes to mutually derive keys and then use the mutually derived keys to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for nodes to mutually derive keys can comprise (i) a key exchange 411a step for SE 102 to derive a symmetric encryption key 442a and (ii) a key exchange 411b step for a configuration server 112 to derive the same symmetric encryption key 442a. Exemplary steps for nodes to encrypt and decrypt data can comprise (i) an encryption 412a step for SE 102 to utilize the symmetric encryption key 442a to convert plaintext to ciphertext, and (ii) a decryption 412b step for configuration server 112 to utilize the symmetric encryption key 442a to convert the ciphertext received from SE 102 into plaintext. The use of all steps a key exchange 411a, a key exchange 411b, encryption 412a, and decryption 412b were also depicted and described in connection with Figure 4a above. Additional detail regarding the use of these steps will be provided herein. Further, although the steps are depicted specifically for the use of particular keys and plaintext/ciphertext combinations in Figure 4b, the steps illustrated can be used with different keys and plaintext/ciphertext combinations. For example, Figure 4c below depicts the same functionality used by the nodes as in Figure 4b, but with (i) different PKI keys input to mutually derive a different symmetric encryption key, and (ii) different plaintext encrypted and different ciphertext decrypted.

A key exchange 411a step for SE 102 to derive a symmetric encryption key 442a can utilize a set of cryptographic algorithms 141 as depicted and described in connection with Figure 1b above. As depicted in Figure 4b, an SE 102 can input both of ephemeral public key from configuration server PKe.CS 402a and initial secret key SK0.SE 102y into a key exchange algorithm 447. The key exchange algorithm 447 could comprise a Diffie Hellman key exchange (DH), an Elliptic Curve Diffie Hellman key exchange (ECDH), and other possibilities exist as well without departing from the scope of the present invention. For some exemplary embodiments, the key exchange algorithm 447 could comprise a

supersingular isogeny Diffie-Hellman key exchange (SIDH). A key exchange algorithm 447 can support either PKI keys based on elliptic curves or RSA algorithms, although support of elliptic curves may be preferred in some exemplary embodiments due to their shorter key lengths and lower processing requirements. A summary of ECDH is included in the Wikipedia article titled "Elliptic Curve Diffie-Hellman" from March 9, 2018, which is herein incorporated by reference.

An exemplary embodiment of key exchange algorithm 447 could comprise a "One-Pass Diffie-Hellman, C(1, 1, ECC CDH)" algorithm as described in section 6.2.2.2 on page 81 of the National Institute of Standards and Technology (NIST) document "NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" from March, 2007 which is hereby incorporated by reference in its entirety. Other key exchange algorithms in NIST SP 800-56A could be utilized as well for a key exchange algorithm 447 in Figure 4b and Figure 4c without departing from the scope of the present invention. For some exemplary embodiments, a key exchange 447 could comprise a key encapsulation mechanism (KEM), where a node such as configuration server 112 encrypts data such as key 442a with a public key such as PK0.SE 102n, and then key 442a can be decrypted with SK0.SE 102y. The ciphertext of key 442a could be sent in message 403.

Other algorithms to derive a shared symmetric encryption key 442a using public keys and private keys may also be utilized in a key exchange algorithm, such as, but not limited to, the American National Standards Institute (ANSI) standard X-9.63. Cryptographic parameters 407 can also include information for using a key exchange algorithm 447 in order to derive a commonly shared symmetric encryption key 442a. Parameters 407 input into a key exchange algorithm 447 can include the key length in bits, an elliptic curve utilized for ECC, a time-to-live for a key 442a that is derived, and similar settings. Additional cryptographic parameters 407 for a PKI keys input into key exchange algorithm 447 can include a supported point formats extension, where the supported point formats extension could comprise uncompressed, compressed prime, or "compressed char2" formats, as specified in ANSI X-9.62. In other words, an ECC keys input into a key exchange algorithm 447 may have several different formats and a set of parameters 407 can be useful to specify the format.

As depicted in Figure 4b, the output of a key derivation function 447 such as an ECDH key exchange can be input into a key derivation function 448. Key derivation function 448 can process the output of a key exchange algorithm 447 in order to further obfuscate and optimize a derived shared secret from the key exchange algorithm 447. For example, the number of bits output from a key exchange algorithm 447 (such as an

exemplary 256 bits) may not match the number of bits utilized for a symmetric encryption key 442a (such as an exemplary 192 bits), and consequently a key derivation function 448 could reduce, process, or truncate the output of a key exchange algorithm 447.

5 In exemplary embodiments, key derivation function 448 could also utilize a secure hash algorithm such as, but not limited to SHA-256, SHA3, or similar algorithms operating on the output of key exchange algorithm 447 in order to make a resulting symmetric encryption key 442a more secure. Further, key derivation function 448 could utilize multiple rounds of XOR logic or processing on the sequence of bits output from the key derivation function 447. Parameter 407 can specify values for the operation of key
10 derivation function 448 such as the number of bits for keys, a hash algorithm to utilize, the number of rounds of XOR on a sequence of bits, and other values as well. As depicted in Figure 4b, the output of a key derivation function 448 can be both a symmetric encryption key 442a and a MAC key 441a, where MAC key 441a can function to verify message integrity of ciphertexts generated with the symmetric encryption key 442a.

15 As depicted in Figure 4b, an encryption 412a step could be utilized by SE 102 with the derived symmetric encryption key 442a in order to convert plaintext 443 into ciphertext 415. An encryption 412 step can utilize a symmetric ciphering algorithm 444a, where symmetric ciphering algorithm 444a could comprise an algorithm according to the Advanced Encryption Standard (AES), Blowfish, and Triple Data Encryption Standard
20 (3DES), and other possibilities exist as well. Parameters 407 could specify the algorithm for a symmetric ciphering algorithm 444a as well as key length and a block cipher mode such as ECB, CBC, OFB, and CFB. Parameters 407 may also specify an authenticated encryption mode such as CCM (NIST SP800-38C), GCM (NIST SP800-38D), and other examples exist as well. Symmetric encryption algorithm 444a can accept input of (i)
25 plaintext 443, (ii) symmetric encryption key 442a, (iii) parameters 407, and (iv) initialization vector (IV) 445a, and output ciphertext 415. As depicted in Figure 4b, and encryption 412a step can output a message authentication code (MAC) of 446a, where the MAC 446a was calculated with the MAC key 441a. MAC 446a allows the other party to verify message integrity of ciphertext 415 by calculating the same MAC 446a value using
30 the MAC key 441a. IV 445a can comprise a random number to introduce randomness into the ciphertext 415, and IV 445a can be transmitted as a plaintext along with the ciphertext 415 in exemplary messages, such as message 413 that contains ciphertext1.SE 415 as depicted in Figure 4a above.

Key derivation 411b step for configuration server 112 depicted in Figure 4b can
35 correspond to key derivation 411a step by SE 102. Key derivation 411b step can comprise the use of ephemeral secret key SK_e.CS 402b and the secure element public key PK₀.SE

102n. Other elements or algorithms within a key derivation 411b step can be equivalent to a key derivation 411a step above, including the use of shared parameters 407. In this manner, configuration server 112 and SE 102 can securely derive the same symmetric encryption key 442a as depicted in Figure 4b. A decryption 412b step for configuration
5 server 112 allows configuration server 112 to convert the ciphertext 415 received into plaintext 443. Decryption 412b step can utilize a symmetric decryption algorithm 444b, which could comprise the same algorithm used in symmetric encryption algorithm 444a except the algorithm being used for decryption instead of encryption. For example, if symmetric encryption algorithm 444a in Figure 4b comprises an AES-192 encryption, and
10 symmetric decryption algorithm 444b can comprise an AES-192 decryption. Note that the same values are input into symmetric decryption algorithm 444b as symmetric encryption algorithm 444a, such as symmetric encryption key 442a, parameters 407, and IV 445a in order to convert ciphertext 415 back into plaintext 443. Configuration server 112 can then read and process plaintext 443 after a decryption 412b step. Configuration server 112 can also verify MAC code 446a using MAC key 441a from a step 411b in order to verify the
15 integrity of ciphertext 415.

Figure 4c

Figure 4c is a series of flow charts illustrating exemplary steps for nodes to
20 mutually derive keys and then use the mutually derived keys to encrypt and decrypt data, in accordance with exemplary embodiments. Exemplary steps for nodes to mutually derive keys can comprise (i) a key exchange 421a step for configuration server 112 to derive a symmetric encryption key 442b and (ii) a key exchange 421b step for SE 102 to derive the same symmetric encryption key 442b. Exemplary steps for nodes to encrypt and decrypt
25 data can comprise (i) an encryption 424a step for configuration server 112 to utilize the symmetric encryption key 442b to convert plaintext to ciphertext 427, and (ii) a decryption 424b step for SE 102 to utilize the symmetric encryption key 442b to convert the ciphertext 427 received from configuration server 112 into plaintext. The use of all steps for a key exchange 421a, a key exchange 421b, encryption 424a, and decryption 424b were also
30 depicted and described in connection with Figure 4a above. Further, although the steps are depicted specifically for the use of particular keys and plaintext/ciphertext combinations in Figure 4c, the steps illustrated can be used with different keys and plaintext/ciphertext combinations. For example, Figure 4b above depicts the same functionality use by the nodes as in Figure 4c, but with (i) different PKI keys input to mutually derive a different
35 symmetric encryption key, and (ii) different plaintext encrypted and different ciphertext decrypted.

As depicted in Figure 4c, a key exchange 421a step can comprise configuration server 112 using ephemeral secret key SKe.CS 402b' and a new, updated SE 102 public key PK1.SE 410a with key exchange algorithm 447. SKe.CS 402b' could be derived in a step 402b' using parameters 407'. As described above in connection with Figure 4a, the output of key exchange algorithm 447 can be input into key derivation function 448. The output of key derivation function 448 can be both a new, second symmetric encryption key 442b and a new, second MAC key 441b. Parameters 407 can be the same for a key exchange 421a and a key exchange 411a in exemplary embodiments, but the parameters 407 could also be different. For embodiments where parameters 407 are the same as a key exchange 411a, then SKe.CSS 402b could be used in step 421a. For embodiments with a different set of parameters 407', then a second SKe.CS 402b' from a step 402' could be used with step 421a. For example, parameters 407' in key exchange 421a could specify longer key length for key 442b compared to key 442a (such as an exemplary 256 bits instead of an exemplary 192 bits), and in this case a key derivation function 448 in key exchange 421a and key exchange 421b could output a longer symmetric encryption key 442b with an exemplary length of 256 bits.

Other possibilities exist as well for parameters 407 to change between a key exchange 421a and a key exchange 411a without departing from the scope of the present invention. A key exchange 421b step can comprise SE 102 using the derived secret or private key SK1.SE 410b with the ephemeral public key from configuration server of PKe.CS 402a' in order to derive the same symmetric encryption key 442b, using the same parameters 407' as key exchange 421a step. The same MAC key 441b can be derived in both a key exchange 421a and a key exchange 421b.

As depicted in Figure 4c, an encryption 424a step for configuration server 112 can utilize the new, second symmetric encryption key 442b for input into the symmetric encryption algorithm 444a. The plaintext may comprise data depicted with message 425 in Figure 4a, and the resulting ciphertext can comprise ciphertext1.CS 427. The plaintext for an encryption step 424a can comprise a device configuration. MAC key 441b can be used to calculate MAC 446b in order to check and confirm validity and integrity of the transmission and proper receipt of ciphertext1.CS 427, such as the presence of any bit errors from the transmission for ciphertext1.CS 427 from configuration server 112 to SE 102. In exemplary embodiments, a new, second initialization vector 445b is created by configuration server 112 and utilized with ciphertext1.CS 427. A decryption 424b step for SE 102 can utilize the new, second symmetric encryption key 442b from step 421b for input into the symmetric decryption algorithm 444b. The ciphertext ciphertext1.CS 427 can be input by SE 102 into the symmetric decryption algorithm 444b along with the

symmetric encryption key 442b in order to output and read the plaintext values. SE 102 can also verify the integrity of ciphertext1.CS 427 by computing a MAC value 446b using the MAC key 441b with the plaintext and a MAC algorithm to confirm the computed MAC value 446b matches the received MAC value 446b, where equal values for the computed and received MAC value 446b means that the integrity of received ciphertext1.CS 427 is confirmed.

Figure 4d

Figure 4d is an illustration of a certificate for a device public key, where the device derived the public key and a corresponding private key, in accordance with exemplary embodiments. Public and private keys in system 100 can utilize PKI techniques based on either RSA or ECC algorithms. A certificate cert1.SE 419 based on an ECC public key for SE 102, such as PK1.SE 410a is illustrated in Figure 4c, although RSA keys or other algorithms supported by post-quantum cryptography could be utilized instead, such as lattice-based cryptography, code-based cryptography, etc. One benefit of using ECC is that an equivalent level of security can be obtained for a much smaller key length. Also, energy may be conserved using ECC algorithms compared to RSA algorithms. Smaller key lengths save bandwidth, memory, processing resources, and power, which are all valuable for a device 101 to conserve a battery and usage of radio-frequency spectrum. For example, an ECC key length of 283 bits can provide security similar to an RSA key length of approximately 2048 bits. Public key PK1.SE 410a can comprise an ECC key in an X.509 certificate, as illustrated in Figure 4c. The values to determine an elliptic curve defining equation could be stored in parameters 407 or 407', and the defining equation or curve name utilized with the key can be disclosed in the certificate. In addition, although the certificate is depicted as cert1.SE 419, the certificate can be used to authenticate device 101 since SE 102 and the identity 102m in cert1.SE 419 can be uniquely and permanently bound to device 101. In other exemplary embodiments, the device identity 101b can be used instead of or in conjunction with ID.SE 102m in a cert1.SE 419, and in those embodiments the device identity 101b could be directly inside a certificate cert1.SE 419.

Certificate 419 could include a signature 220, where signature 220 can be signed using ECC signature techniques, such as the Elliptic Curve Digital Signature Algorithm (ECDSA) for signature algorithm 227 with a secure hash such as SHA256 for a message digest 230. The creation of a signature 220 is depicted and described in connection with Figure 2b above. In order to generate signature 220, the private key associated with certificate authority 115 may also be an ECC-based private key using similar parameters

407. Note that the parameters 407 for a parent public key/certificate do not have to be identical, and the parent key/certificate belonging to CA 115 could have a longer validity or different key lengths, as examples. Note that the public key PK1.SE 410a in a certificate 419 could be based on different algorithm or curve than the private key used for signature 220 in certificate 419. For example, that the public key 410a can be an ECC key, while the signature 220 in certificate 419 could be generated with RSA algorithm and key. Certificate 419 may also include parameters 407', where parameters 407' can specify an elliptic curve utilized with the public key PK1.SE 410a. Parameters 407' could also include the start and end times for the validity of either public key PK1.SE 410a or certificate 419. Other parameters 407' can be utilized in a certificate 419 as well, and parameters 407' may specify values that are not included or external to a certificate 419.

Certificate 419 illustrated in Figure 4c illustrates exemplary embodiments of the present invention. Over the lifetime of a device 101, which could be a decade or longer, multiple device public keys such as PK1.SE 410a may be utilized. In other words, future public keys for device 101 could include a PK2.SE, PK3.SE, etc., and these keys could be signed by a certificate authority 115. The potential use of multiple different module public keys such as key 410a include (i) the expiration of a certificate 419 (including expiration of a public key associated with a certificate authority used in signature 220), (ii) a need to change an elliptic curve specified in a parameters 407', (iii) adding a new public/private key pair for connection with a different reporting system 120, (iv) increasing a key length utilized in a public/private key pair, (v) the transfer of ownership or control of device 101, where a new or different device owner 122 may require or prefer that keys with devices be rotated or renewed, and/or (vi) device 101 connecting to a new server that utilizes a different cryptographic algorithms (i.e. RSA-based instead of ECC-based). Other possibilities exist as well for reasons an SE 102 to derive a new public key PK1.SE 410a. Further, although only one public key PK1.SE 410a is depicted in Figure 4a and Figure 4c, a device 101 may utilize multiple public keys concurrently, and thus required multiple different certificates 419 for each of the different public keys. In exemplary embodiments (i) a first device public key PK1.SE 410a in a first certificate 419 can be utilized with a first reporting system 120, and (ii) a second device public key PK1.SE 410a (possibly derived using a different set of parameters 407' including using a different elliptic curve or different cryptographic algorithm) with a second certificate 419 can be utilized with a second reporting system 120 and/or access network 126.

As illustrated in Figure 4c, an identity associated with device 101 could be included in the "Common Name" (CN) field, which is depicted as ID.SE 102m. The identity associated with device 101 could be ID.device 101b and used instead of ID.SE 102m in a

certificate 419. Either ID.device 101b or ID.SE 102m could be recorded with a “Serial Number” (SN) field in the certificate. Alternatively, ID.SE 102m or ID.device 101b could be recorded in the “Organizational Unit” (OU) field, and other locations for storing or recording identities within a certificate as illustrated in Figure 4c exist as well. In addition, a specific identity or sequence count or sequence number for the public key of SE 102 and device 101 could be recorded in a certificate 419, so that PK0.SE 102n or PK0.SE 102n’ can be differentiated from PK1.SE 410a.

Also, as noted previously herein, the use of a certificate 419 may optionally be omitted, such that device 101 and reporting system 120 or configuration system 114 share public keys without using certificates 419, or the certificates could take a form different than that depicted in Figure 4c. A signature portion 220 of a certificate 419 can also include a value for “r” or similar value used with a non-deterministic use of a digital signature algorithm 227, where the value of “r” was described in Figure 2b above. In other words, an “r” value in a signature 220 of a certificate 419 or other ECDSA certificates described herein can comprise (i) a random number used once in a non-deterministic mode, or (ii) a fixed, deterministic value based on a message digest 230 in Figure 2b which could comprise a deterministic mode. Further, in exemplary embodiments, certificate 419 with a certificate authority signature 220 can comply with IETF RFC 5480 standard “Elliptic Curve Cryptography Subject Public Key Information”.

Figure 5

Figure 5 is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a mobile handset and a device, in accordance with exemplary embodiments. Before initiating steps and message flows depicted in Figure 5, mobile handset 108, device 101, and the other elements depicted for system 500 in Figure 5 may previously complete exemplary message flows and steps depicted in Figure 3 and Figure 4a above. System 500 can include a configuration server 112, mobile handset 108, and device 101. An exemplary difference between system 300 depicted in Figure 4a and system 500 depicted in Figure 5 can be the use of a Wi-Fi access point 501 for mobile handset 108 and a Wi-Fi client 502 for device 101.

In a system 500, mobile handset 108 and configuration server 112 can continue secure session 321 from Figure 4a above, which was also established in Figure 3 above. If a first secure session 321 terminates, such as device 101 is in a location without connectivity to an access network 126 when NFC session 303 is established, then mobile handset 108 could move “back and forth” or between (i) a first location with device 101 and (ii) a second location with communication to access network 126 for communication

with configuration server 112. In this embodiment, secure session 321 may (i) close as mobile handset 108 moves to the location with device 101 and (ii) reopen as mobile handset 108 moves to the location with communication to access network 126 for mobile handset 108.

5 As depicted in Figure 5, mobile handset 108 can conduct a Wi-Fi access point activation step 437b. Configuration data for mobile handset 108 to utilize with Wi-Fi access point 501 could be included in data Wi-Fi-AP.MH 437a from configuration server 112. Values or data in Wi-Fi-AP.MH 437a could comprise frequencies or channels to utilize, a service set identifier (SSID) or network name, user identities and passwords,
10 and/or public certificates for client 502 and access point 501, etc. In exemplary embodiments, Wi-Fi access point 501 does not broadcast the SSID such as in a broadcast frame, and the only client allowed can be device 101 using client 502. In addition, by using an obfuscated and non-broadcasted SSID (possibly a pseudo random string) that has only been shared outside mobile handset 108 with device 101 via ciphertext 441 with Wi-
15 Fi-AP.MH 437a, then only device 101 could reasonably connect with Wi-Fi access point 501. Further, an access list of allowed users for Wi-Fi access point 501 can result in only device 101 being able to connect via Wi-Fi connection setup 503 in Figure 5. Further, Wi-Fi connection setup 503 can utilize standard Wi-Fi security protocols such as, but not limited to, WPA2, WPA3, or subsequent standards. In exemplary embodiments, Wi-Fi
20 connection setup 503 utilizes a preshared secret key (PSK), where the PSK is received by mobile handset 108 in message 438 in Figure 4a above and sent to device 101 in message 440 in Figure 4a. As depicted in Figure 5, in exemplary embodiments, mobile handset 108 and device 101 can utilize network credentials of “user name, password 437aa” to secure connection setup 503.

25 In exemplary embodiments, Wi-Fi-AP.MH 437a can specify a low maximum output power for Wi-Fi access point 501, such as an exemplary 1 milliwatt or less. Wi-Fi-AP.MH 437a can also specify the Wi-Fi standard to utilize within the family of 802.11 standards from IEEE, such as version g, n, ac, etc. During use of a Wi-Fi access point 501 as depicted with system 500, device 101 may normally be in close physical proximity of
30 mobile handset 108, such as less than an exemplary few meters, and consequently relatively lower transmit power for mobile handset 108 could be utilized, compared to the typical maximum power typical for standard Wi-Fi usage in 2018 of approximately 200 milliwatts. In addition, although a Wi-Fi access point 501 is depicted in Figure 5, mobile handset 108 and device 101 could implement a Bluetooth network, where either device 101
35 or mobile handset 108 can function as the master. For this embodiment, parameters and credentials for a Bluetooth network for connection setup 503 could be passed in the data

Wi-Fi-AP.MH 437a. Other short-range and LAN wireless network protocols besides Wi-Fi and Bluetooth could be utilized as well without departing from the scope of the present invention. A configuration database 112a could record if device 101 should utilize Wi-Fi, Bluetooth, or another wireless protocol for connection setup 503. As mentioned above, in
5 exemplary embodiments where the size of software package 104q is deemed to be sufficiently small to be supported by NFC session 303, then a separate connection setup 503 could be omitted and the NFC session 303 could be utilized instead for a system 500 in Figure 5.

For device 101 in system 500, device 101 could use a step 503a in order to activate
10 a Wi-Fi client 502. Parameters and credentials for Wi-Fi client 502 could be received in a message 440 above with Wi-Fi-AP.MH 437a. For a step 503, SE 102 which received data in Wi-Fi-AP.MH 437a could send (i) data to device 101 CPU 102c and radio 101z via (ii) data bus 101d as depicted in Figure 5 in order to activate Wi-Fi client 502. In exemplary
15 embodiments, the radio 101z for Wi-Fi client 502 in device 101 resides outside of SE 102 as depicted in Figure 1b. However, other embodiments and configurations for SE 102 and radio 101z are possible and also depicted and described below in connection with Figure 6a and 6b. Consequently, other configurations of internal elements such as SE 102 and Wi-Fi
client 502 are possible without departing from the scope of the present invention. A step 503 could include the use of a firewall for SE 102 and/or device 101, such that SE 102
20 would not normally respond to packets, queries, data, or probes via connection 503 unless the data successfully includes a properly decrypted ciphertext3.CS 506 below.

At step 504 in Figure 5, configuration server 112 can prepare software package 104q for transfer to device 101 via mobile handset 108. An exemplary software package 104q is depicted in Figure 5, and was also described above in Figure 4a. Software package
25 104q could also be referred to as a second device configuration, where a first device configuration could be the plaintext in a step 424a above. Configuration server 112 could use a configuration database 112a, an identities list 320, data from device owner 122 and device manufacturer 101x, and data from reporting system 120 in order to assemble software package 104q. Configuration server 112 or configuration system 114 could
30 include an updated secure element firmware 102u' in software package 104q. Software package 104q can comprise a compressed format such as using gzip or other compression techniques. Configuration server 112 can also create a full file list 507 for software package 104q in a step 504. The file list 507 can also include a signature 220 from configuration server 112 using a secret key SK.CS, such that SE 102 can use a signature
35 verification step 221 with the PK.CS from cert.CS 112c. In this manner, SE 102 can verify that software package is from previously authenticated configuration server 112.

In preferred exemplary embodiments, configuration server 112 also conducts a signature verification 221 step on individual software components for software package 104q that configuration server 112 receives from other nodes in a system 100. For example, before (a) configuration server 112 signs software package 104q for device 101, which could include files for an updated device OS 101e where the files for the updated device 101e were received by a device manufacturer 101x, (b) configuration server 112 could conduct a signature verification 221 step for a signature by device manufacturer 101x on the files for updated device OS 101e. In other words, in exemplary embodiments, configuration server 112 verifies signatures on files received for software package 104q before configuration server 112 conducts a signature creation step 220 for the software package 104q. In this manner, elements for software package 104 may be sourced from multiple different parties in a system 100, and configuration server 112 could verify authenticity for each element and provide an overall signature or “stamp of approval” of software package 104q for device 101. In this manner, device 101 may not need to check the signatures on individual elements within software package 104q, which could be difficult if device 101 or SE 102 does not have all the intermediate and root certificates for each of the individual element signatures in a software package 104q.

Configuration server 112 can then conduct an encryption step 424g to create ciphertext3.CS 506. The encryption step 424g can be equivalent to encryption step 424a depicted and described in connection with Figure 4c, where the plaintext to be encrypted can comprise both file list 507 and software package 104q (where software package 104q may be previously compressed). Note that second symmetric encryption key 442b can continue to be used with encryption step 424g. In exemplary embodiments, a different, updated value for initialization vector 445b is used with an encryption step 424g. Configuration server 112 can then use connection 321 to send a message 505 to mobile handset 108, where message 505 includes an ID.transaction4 and ciphertext3.CS 506, where ciphertext3.CS was created in step 424g.

Mobile handset 108 can receive message 505 via connection 321 and forward message 505 to device 101 via connection 503. In exemplary embodiments where device 101 is out of range from an access network 126 utilized by mobile handset 108, mobile handset 108 can receive message 505 and store or cache the data contained such as ciphertext3.CS 506. Mobile handset 108 could then temporarily close connection 321 and move to device 101 and re-establish connection 503, and then subsequently send ciphertext3.CS 506 to device 101. In this manner, secure connection 321 and connection 503 as depicted in Figure 5 do not need to be operating concurrently in exemplary embodiments. The present invention allows device 101 to securely conduct a configuration

step 103 even when device 101 is outside the range of an access network 126 (e.g. device 101 is in a physical location where mobile handset 108 cannot reach an access network 126 to connect with configuration server 112 when mobile handset 108 has a connection 503 or 303 established).

5 Device 101 can receive message 505 over connection 503. Device 101 using SE 102 can conduct a decryption step 424h for ciphertext3.CS 506, where decryption 424h can be equivalent to decryption 424b depicted and described in connection with Figure 4c above. The symmetric encryption key 442b in a decryption step 424h for SE 102 can be the same symmetric encryption key 442b for a decryption step 424f in Figure 4a above.
10 Note that in exemplary embodiments, the same symmetric encryption key 442b can be used by SE 102 to (i) decrypt ciphertext2.CS 441 received through NFC session 303 and (ii) decrypt ciphertext3.CS 506 received a different radio link than ciphertext2.CS 441. SE 102 can use decryption 424h to read the plaintext values of file list 507 and software package 104q.

15 SE 102 can then conduct a step 507, where step 507 can include both (i) SE 102 decompressing software package 104q using an exemplary library such as gzip, and (ii) SE 102 conducting a signature verification 221 step over software package 104q, where software package 104q was previously signed by configuration server 112 in a step 504 above. In this manner, SE 102 and device 101 can be reasonably assured that files and
20 components in software package 104q are genuine or authenticated/approved by configuration server 112. SE 102 can (A) report an error or (B) refuse to process software package 104q or elements within software package 104q if (C) a signature verification step 221 fails. Step 507 can also comprise SE 102 verifying that software package 104q contains all the data and files listed in a file list 507

25 At step 508 in Figure 5, SE 102 and device 101 can record the data within software package 104q in either SE storage memory 102d or device storage memory 101f. In exemplary embodiments, files for device OS 101e can be recorded in device storage memory 101f and files for SE firmware can be recorded in SE storage memory 102d, and other possibilities exist as well. A step 508 can also comprise SE 102 and device 101
30 backing up or storing the running configuration for each before loading data in a software package 104q, and in this manner the system can be restored if applying the updated files for software package 104q fails. After internally storing or loading the files for software package 104q, device 101 and SE 102 can perform a reboot, so that both restart with the new files from software package 104q. Upon a reboot in a step 508, connection 503 may
35 temporarily terminate with the reboot, but device 101 can re-establish connection 503 using the radio 101z and Wi-Fi client 502. A step 508 can then also comprise device 101 and/or

SE 102 creating a report 508a, where report 508a includes a status code with success or errors for each file in file list 507 for software package 104q. In other words, report 508a can record the success or errors of applying each of the files in software package 104q, which may be useful for configuration server 112 or other authorized elements in system 100 to know the state of device 101.

SE 102 can then conduct an encryption step 424i, which could correspond to encryption step 424a depicted in Figure 4c. Encryption step 424i could comprise using report 508a as plaintext and the symmetric encryption key 442b in order to create ciphertext3.SE 510. The symmetric encryption key 442b in an encryption step 424i for SE 102 can be the same symmetric encryption key 442b for a decryption step 424f in Figure 4a above. MAC codes 446b and initialization vectors 445b can also be processed with encryption step 424i and also all steps 424 where encryption or decryption takes place. SE 102 can then send mobile handset 108 a message 509 over connection 503, where message 509 includes the transaction identity ID.transaction4 and the ciphertext3.SE 510. As depicted in Figure 5, mobile handset 108 can then forward the data received in a message 509 over connection 321 to the configuration server 112. Configuration server 112 can then conduct a decryption step 424j in order to convert ciphertext3.SE 510 into plaintext, and thereby read the report 508a as plaintext. Decryption 424j can be equivalent to decryption 424b depicted and described in connection with Figure 4c above. The symmetric encryption key 442b in a decryption step 424j for configuration server 112 can be the same symmetric encryption key 442b for an encryption step 424g above in Figure 5.

Upon reading the plaintext report 508a, configuration server 112 can conduct a step 511 as depicted in Figure 5. Configuration server 112 could record in a configuration server database 112b each of the files in a file list 507 for software package 104q that were successfully installed. For embodiments where some non-fatal errors or warnings were recorded in report 508a, then codes or logs of errors in a report 508a could also be recorded in a configuration server database 112b. A step 511 can also comprise configuration server 112 sending messages to other elements in system 100 in Figure 1a regarding the status of device 101, such as sending (i) device owner 122 a notice of successful load of device reporting application 101v from software package 104q, (ii) reporting system 120 a notice that device 101 should begin reporting, and (iii) access network 126 that device 101 will begin connecting with network access credentials 126a, and other possibilities exist as well for configuration server 112 to use or process data received in a report 508a from device 101 and SE 102. If error codes are for the installation of software are received in a report 508a, such as some software in package 104a not being successfully installed, then configuration server 112 could use a step 511 with a configuration database 112a or 112b

to determine next steps, such as potentially re-sending message 505 or resending a portion of the software package 104q with error codes or conditions identified in a report 508a.

After processing step 511, configuration server 112 could perform a step 512, which can comprise a series of steps to close communications for mobile handset 108 and SE 102. Configuration server 112 could generate a command 512a for mobile handset 108, where command 512a instructs mobile handset 108 to close connection 321. Configuration server 112 could generate a command 512b for device 101, where command 512b instructs device 101 to close connection 503 and also being reporting through reporting system 120 using access network 126 in Figure 1a. Configuration server 112 could generate a configuration user report 512c, where report 512c could be for the configuration user 108a operating mobile handset 108. Report 512c could provide a human readable status for display on mobile handset 108 regarding the status of device 101 and a configuration step 103. Report 512c can also provide manual instructions for any modifications of transducers 105 or manual changes for configuration user 108a to perform for device 101, monitored unit 106, or connected equipment in order to complete the configuration step 103. In an exemplary embodiment, a report 512c could request that configuration user 108a attach an external antenna to device 101 in order to improve RF signal strength with access network 126. In another exemplary embodiment, report 512c could instruct configuration user 108a to perform a manual power cycle of device 101 or monitored unit 106. Other possibilities exist as well regarding instructions for a configuration user 108a in a report 512c to complete a configuration step 103 without departing from the scope of the present invention.

After completing a step 512, configuration server 112 can send mobile handset 108 a message 513 through connection 321, where message 512 can include (i) command 512a for mobile handset 108 to close connection 321, (ii) command 512b for device 101 to close connection 503 and begin reporting through reporting system 120 using access network 126 and credentials 126a, and (iii) configuration user report 512c for configuration user 108a. Mobile handset 108 can receive message 512 and forward the portions of the message 513 for (ii) command 512b to SE 102 in device 101. Mobile handset 108 can then conduct a step 514a to close connection 321 with configuration server 112. Mobile handset 108 can then conduct a step 515, where the configuration user report 512c is displayed to configuration user 108a, and a step 515 can also comprise configuration user 108a entering information in a display for mobile handset 108 confirming any manual changes for device 101 or monitored unit 106. In some embodiments, a step 515 can take place before a step 514a, such that mobile handset 108 can send information back to configuration server 112

and a configuration database 112a regarding any manual changes performed as requested in the configuration user report 512c.

At step 516, mobile handset 108 can then restore the previous Wi-Fi access point credentials for Wi-Fi access point 501 recorded in a step 435 above in Figure 4a. In other words, the original Wi-Fi access point credentials used by mobile handset 108 before a configuration step 103 and connection 503 would preferably be restored, to reduce interruption of services for configuration user 108a. For example, configuration user 108a could have a laptop or tablet that periodically connects with mobile handset 108 using a Wi-Fi access point 502 and credentials that were recorded in step 435. If those credentials from step 435 were not restored for mobile handset 108 in a step 516, then credentials in Wi-Fi-AP.MH 437a would continue to be used and the exemplary laptop or tablet would not normally be able to connect with mobile handset 108.

SE 102 and device 101 can receive the portions of message 513 as depicted in Figure 5, where a portion could comprise command 512b for device 101 to close connection 503. SE 102 and device 101 could then perform a step 514b and close connection 503. In exemplary embodiments, device 101 then loads network access credentials 126a for connecting with access network 126. Device 101 could then select an access network 126 for conducting a connection setup 518. Device 101 could select the access network based upon network access credentials 126a or from a networks available list 322 received in an identity list 320 from a message 425. Other possibilities exist as well for device 101 to select an access network for a connection setup 518 without departing from the scope of the present invention.

Device 101 then conducts a connection setup 518 with access network 126. For a connection setup 518, device 101 can use either (i) the network access credentials 126a or (ii) the private key SK1.SE 410b with cert1.SE 419. For example, with 5G networks that support EAP-TLS, then with connection setup 518, device 101 can (i) authenticate with an access network 126 using SK1.SE 410b and cert1.SE 419 and also (ii) mutually derive a symmetric ciphering key with access network 126 using SK1.SE 410b and cert1.SE 419. In some exemplary embodiments, network access credentials 126a could comprise a profile for an embedded universal integrated circuit card (eUICC). In addition, network access credentials 126a could comprise a pointer, URL, or another identifier of an eUICC profile for device 101 to load. The transfer of network access credentials 126a from a message 426 in Figure 4a above could comprise a transfer of the pointer, URL, or another identifier for the eUICC profile.

In addition, although Figure 5 depicts a step 514b of closing connection 503 before connecting with access network 126, device 101 could connect with access network 126

before closing connection 503. In this manner of device connecting with access network 126 before closing connection 503, device 101 can report error conditions or problems for connecting to access network 126 to mobile handset 108 and configuration user 108a. At step 517, device 101 can load the network access credentials 126a in order to connect with access network 126. Note in exemplary embodiments, the access network 126 used by device 101 can be different than the access network 126 used by mobile handset 108, although the two access networks 126 could be the same in other embodiments.

After loading network access credentials 126a in a step 517, device 101 can connect with access network 126 in a step 518. Although an SE 102 is depicted as conducting the connection 518, device 101 may perform the connection 518 such as using a radio 101z for device 101 as depicted in Figure 1b. Connection 518 could be through either a wireless LAN such as Wi-Fi, where network access credentials 126a can comprise data equivalent or similar to data in Wi-Fi-AP.MH 437a, or connection 518 would be through a wireless WAN such as any of a 3G, 4G, or 5G wireless network, including a low power wide area network over non-licensed spectrum such as in the 900 MHz ISM band. As depicted in Figure 5, connection 518 can connect device 101 securely with access network 126 using network access credentials 126a.

At step 519, device 101 can process data in order to connect with reporting server 116, where reporting server 116 can be a server or set of servers in a reporting system 120. Data processed in a step 519 can include collecting data from transducers 105 using device reporting application 101v via a transducer bus 101i, which can include startup or initialization data for transducers 105. In exemplary embodiments, a step 519 also includes a calibration of transducers 105 with assistance of configuration user 108a and mobile handset 108. Mobile handset 108 could display to configuration user 108a instructions or steps related to calibration, such as providing a known reference signal to a transducer 105. An example of providing a known reference signal for transducer 105 could be putting a temperature probe 105ba such as a thermocouple or thermistor in an ice water bath, and other examples exist as well. In other exemplary embodiments, a calibration of transducers in a step 519 can be omitted.

SE 102 in device 101 for a step 519 could also verify a certificate for reporting server 116, reading a DNS name for reporting server 116, a TCP or UDP port number to connect with at reporting server 116, and other steps or data for device 101 or SE 102 could be in the config.reporting-system 120a received in a message 425 depicted in Figure 4a. A certificate for reporting system 120 or reporting server 116 could be included in config.reporting system 120a. In exemplary embodiments, config.reporting-system 120a

contains a list of cryptographic parameters to utilize when communicating with reporting system 120 and reporting server 116 (such as a subset or superset of parameters 407).

Device 101 can then conduct a series of steps in order to establish a secure communications link with reporting server 116. Figure 5 depicts an overview of the connection between device 101 and both access network 126 and reporting server 116, and consequently only a subset of the steps and message flows between those elements are depicted in Figure 5. In exemplary embodiments, the connection between device 101 and reporting server 116 comprises a “datagram TLS” (DTLS) connection as specified in IETF RFC 6347, although other protocols could be used as well. In exemplary embodiments, device 101 sends reporting server 116 the certificate cert1.SE 419 in a message 520, where cert1.SE 419 was received by device 101 and SE 102 above in a message 425 depicted and described in connection with Figure 4a.

In other words, device 101 can begin communicating with reporting server 116 and reporting system 120 using a certificate based on a PKI key pair that was derived by device 101 and SE 102 during a configuration step 103. Cert1.SE 419 can include PK1.SE 410a, and SE 102 can use SK1.410b in with internal, secure, private operations for data in cert1.SE 419. The previously installed certificate cert0.SE and private key SK0.SE 102y can be deprecated or replaced with the new credentials. In this manner, security for device 101 can be increased, since SK0.SE 102y could physically be in the hands or control of other parties besides device owner 122 before installation or operation with monitored unit 106. Further, preferred security practices may require that secret keys and certificates get rotated over time, especially when validity dates for certificates expire, and thus sending a new certificate cert1.SE 419 to reporting server 116 may be periodically required. An exemplary certificate cert1.SE 419 is depicted in Figure 4c. Further, in other exemplary embodiments, message 520 comprises device 101 sending an identity of device 101 such as ID.device 101b to reporting server 116, and reporting server 116 then uses ID.device 101b to fetch the certificate cert1.SE 419 from configuration system 114 or device owner 122.

In addition, over approximately the next decade, devices deployed with cryptographic algorithms and keys supporting ECC cryptography may need to update to support post-quantum cryptography in order to remain reasonably secure. An installed device initially using ECC cryptography could update private keys and algorithms to new post-quantum cryptography algorithms using the steps disclosed herein, including receiving a cert1.SE 419 supporting post-quantum cryptographic algorithms after a device has been initially deployed supporting only ECC and/or RSA based algorithms.

Reporting server 116 can receive the certificate cert1.SE 419 and then use a step 521a to verify the certificate, including using a signature verification step 221 depicted in

Figure 2b. Step 521a could also comprise reporting server 116 and reporting system 120 verifying that device 101 is authorized to use reporting system 120, after device 101 with device identity 101b has been authenticated. Since message 520a could comprise an initial receipt of data from device 101 using ID.device 101b, step 521a could also comprise
5 reporting system fetching or querying configuration system 114 using ID.device 101b for configuration data pertaining to device 101. The device configuration data for reporting server 116 received in a step 521a could be stored in the configuration database 112a or 112b, and can include identity list 320, networks available list 322, software package 104q, files list 507, authentication parameters 111d, and parameters 407. With data from a step
10 521a, reporting server 116 can select subsequent steps or procedures in order to communicate with device 101 and SE 102 inside device 101.

Step 521b for reporting server 116 can comprise reporting server 116 generating an ephemeral public key PKe.RS 522 and an associated ephemeral secret key SKe.RS. For a step 521b, reporting server 116 could use a similar set of steps described in step 402 above
15 for configuration server 112 to derive a PKI key pair, including the use of a random number generator, a random number, a key pair generation function, and a set of parameters 407. Further, although Figure 5 depicts reporting server generating and sending an ephemeral key PKe.RS 522, in exemplary embodiments, SE 102 can (i) generate the ephemeral key pair using similar steps as described above for reporting server 116 or
20 configuration server 112 in a step 402, and (ii) send the signed ephemeral public key to reporting server 116. Step 524a for reporting server 116 can comprise reporting server 116 using a key exchange 421a step as depicted in Figure 4c, except reporting server 116 would utilize PKe.RS 522 as the ephemeral key input into key exchange algorithm 447, along with PK1.SE 410a. The output of step 524a would be a new symmetric encryption key
25 442b for use by reporting server 116 with device 101 and SE 102. Note that a key exchange step 524a could continue to use PK1.SE 410a, which could be received in cert1.SE 419. For the alternative embodiment where the ephemeral public key is derived by SE 102, then a key exchange 524a step for reporting server 116 can comprise reporting server 116 inputting an ephemeral key PKe.SE and a secret key SK.RS into key exchange
30 algorithm 447.

Step 220i for reporting server 116 can comprise reporting server 116 using a signature creation 220i step in order to create signature1.RS 523. The secret key used to create signature1.RS 522 can correspond to the public key in a certificate for reporting server 116. The message to sign for signature1.RS 522 can comprise the ephemeral public
35 key PKe.RS 522, which could be derived in a step 521b above. Reporting server 116 can then send PKe.RS 522 and signature1.RS 523 to SE 102 as depicted in Figure 5. SE 102

can receive the data, and conduct a signature verification step 221i in order to verify signature1.RS 523, using a public key for reporting server 116 in a certificate for reporting server 116. A certificate for reporting server 116 could be received with Config.Reporting System 120a in message 425. At step 524b, SE 102 can conduct a key exchange 524b step which corresponds to the key exchange 524a step conducted by reporting server 116. Step 524b for SE 102 can comprise SE 102 using a key exchange 421b step as depicted in Figure 4c, except SE 102 would utilize PKe.RS 522 as the ephemeral key input into key exchange algorithm 447. The output of step 524b would be a new symmetric encryption key 442b for use by SE 102 and device 101 with reporting system 116.

Device 101 can then conduct at step 525, which can comprise device 101 collecting or sending transducer data 125 for transducers 105. A step 525 could also comprise device 101 running through a set of configuration test vectors 104x and generating a report 525a. Configuration test vectors 104x could comprise device 101 testing dynamic range of transducers 105, having configuration user 108a provide test data for transducers 105, or also having transducers 105 collect initial, "production" data regarding monitored unit 106. Report 525a can comprise a set of data generated by device 101 using transducers 105 with monitored unit 106, including via a configuration test vector 104x. In other exemplary embodiments, a configuration test vector 104x could be omitted and transducers 105 and device 101 could simply begin operation with monitored unit and data could be recorded in a report 525a.

Device 101 using SE 102 can then conduct an encryption step 424k as depicted in Figure 5, and SE 102 can conduct an encryption step 424k to create ciphertext4.SE 527. The encryption step 424k can be equivalent to encryption step 424a depicted and described in connection with Figure 4c, where the plaintext to be encrypted can comprise transducer data 125 and/or report 525a. Note that symmetric encryption key 442b computed in step 524b can be used with encryption step 424k, which can also be equal to the symmetric encryption key 442b computed in step 424m below for reporting server 112. Device 101 can then send reporting server 116 a message 526, where message 526 can include ciphertext4.SE 527, where the plaintext for ciphertext4.SE 527 can comprise transducer data 125 and report 525a.

As depicted in Figure 5, reporting server 116 can receive message 526 and conduct a decryption step 424m. Decryption step 424m can utilize the symmetric encryption key 442b derived in a step 524a above, and decryption step 424m can be equivalent to decryption step 424b depicted in Figure 4c. After decryption step 424m reporting server 116 can read the plaintext values of transducer data 125 and report 525a. At step 528, reporting server 116 can record plaintext data from message 526 in a reporting database

118 or also send data to device owner 122. If expected values for transducer data 125 are received and report 525a meets acceptance criteria, then at step 528 (i) reporting server 116 and reporting system 120 can record that configuration step 103 has been successfully completed, and (ii) reporting system 120 could inform device owner 122 and/or configuration system 114 in Figure 1a that a configuration step 103 of device 101 has been successfully completed. In exemplary embodiments, upon successful completion of a configuration step 103 of SE 102 then SE 102 can be depicted as a secure processing environment 104 or SE 104 as shown in Figure 1a. If error conditions are noted for transducer data 125 or report 525a, then configuration user 108a, device owner 122, and configuration system 114 could be notified by reporting server 116. Upon conclusion of step 528 in Figure 5, both (i) device 101 with SE 104 and (ii) reporting server 116 can proceed with regular operation of device 101 with monitored unit 106 since the configuration step 103 has been completed.

Mobile handset 108 can conduct a step 529, where step 529 can comprise the configuration application 108b sending a query 530 to reporting server 116 in order to confirm successful completing of configuration step 103. As described in the paragraph above, reporting server 116 could determine that configuration step 103 is complete based on the successful receipt of transducer data 125 and report 525a. Receipt of a report 525a can confirm any of: (i) network access credentials 126a for access network 126 function, (ii) device 101 has wireless or wired connectivity properly established, (iii) transducers 105 are connected to device 101 and functioning, (iv) that transducers 105 are collecting transducer data 125 for monitored unit 106, (v) a secure processing environment (SE) 102 within device 101 has been configured with updated keys such as PK1.SE 410a and SK1.SE 410b, (vi) software package 104q has been downloaded and applied for device 101 and SE 102, (vii) a complete set for an identity list 320 has been received, including physical location of device 101 such as geographical coordinates 325, (viii) an updated certificate cert1.SE has been processed and received by reporting system 116, (ix) reporting system 116 can communicate with device 101 in a secure manner, (x) device 101 can utilize a “backup” or second access network 126 with a second set of network access credentials 126a if the first access network 126 becomes unavailable, and (xi) a configuration step 103 for device 101 properly functions, such that a configuration step 103 could be conducted a second time in the future if required, and/or (xii) device 101 records root and intermediate certificates in order to verify a certificate for reporting server 116, configuration server 112, and authentication server 111.

As depicted in Figure 5, reporting server 116 can reply to query 530 with a message 531 of “device OK”, if configuration step 103 has been confirmed by reporting system 116

as completed, such as through any of (i) through (xii) in the paragraph above. Not all of (i) through (xii) are required for some embodiments. In exemplary embodiments, upon successful receipt of a message 531 with a status of “device OK” for device 101 or an equivalent message 531, then mobile handset 108 can display to configuration user 108a that a configuration step 103 for device 101 has been completed and device 101 can be left in operation, potentially without additional steps or manual changes to be performed by configuration user 108a. For embodiments where message 531 provides a status of “device not OK”, then mobile handset 108 could display to configuration user 108a an error status and configuration user 108a could work with device 101 and configuration system 114 in order to diagnose and rectify the error code in order to receive a subsequent message 531 “device OK”.

Figure 6a

Figure 6a is a graphical illustration of components for a device, in accordance with exemplary embodiments. Device 101 as depicted in the above Figure 1a through Figure 5 may comprise several different physical and logical embodiments in the present invention, and Figure 6a depicts a different physical configuration for device 101 than that depicted in Figure 1b. Device 101 as depicted in Figure 6a herein and Figure 6b below can still utilize the components and sub-steps for a configuration step 103 as contemplated herein. As depicted in Figure 6a, a device 101 could include a central processing unit (CPU) 101c, a radio 101z, a transducer bus TR bus 101i, an NFC radio 102c, and a nonvolatile memory storage 101f. The elements for device 101 in Figure 6a can provide equivalent functionality for the elements depicted in Figure 1b above.

As depicted in Figure 6a, a secure processing environment (SE) 102 can be included within CPU 101c in exemplary embodiments. In other words, a separate, dedicated processor for SE 102 is not required in order to obtain the benefits from a configuration step 103 in the present invention, and the functionality of SE 102 could be provided by CPU 101c. In exemplary embodiments, the functionality of SE 102 could either be (i) physically separate hardware within CPU 101c, such as a dedicated processing core or memory registers and transistors that are dedicated to the functionality of SE 102, or (ii) logically separate functions within CPU 101c, such as dedicated “time slices” during the operation of CPU 101c. For a logically separate function of SE 102 within CPU 101c, CPU 101c can switch from a first mode to a second mode, where operation in the first mode can comprise function as a CPU 101c for device 101, and the second mode can comprise a secure processing environment 102. For example, a memory register within SE 102 in Figure 6a could (X) record secret key SK0.SE 102y or SK1.SE 410b in order to

process cryptographic algorithms 141 such as encryption step 412a or 424b when (Y) CPU 101c in Figure 6a operates in the second mode described in the sentence above. At a later time, CPU 101c could switch to the first mode where CPU 101c does not operate as SE 102 and in that case the memory register recording a secret key could be “invisible” or “not available” to CPU 101c. Other possibilities exist as well for a CPU 101c and an SE 102 to be combined without departing from the scope of the present invention.

A radio 101z for device 101 could operate in any mode supporting 2G, 3G, 4G, 5G, or subsequent wireless standards in order to connect with an access network 126. Radio 101z could also support other wireless standards such as the 802.11 “Wi-Fi” standards, or low power wide area networking such as Sigfox, LoRa, or NB-IoT. A device 101 could also use multiple radios 101z in order to connect with different wireless networks. A second radio in device 101 in Figure 6a can comprise NFC 102c, which could utilize standards supporting “near field communications”. The radio NFC 102c can include components similar to radio 101z, but supporting operating at a frequency near 13 MHz and also at low power. Either radio 101z or radio NFC 102c can include standard radio components such as RF filters, RF amplifiers, a clock, and phased loop logic (PLL), and may be connected with an antenna. As described above, although radio 102c is depicted as “NFC 102c”, in other exemplary embodiments, radio 102c can operate using wireless standards other than “near field communications”. In exemplary embodiments radio 102c could operate as a Bluetooth radio (e.g. IEEE 802.15.1 or subsequent), or a Wi-Fi radio supporting standards within the family of standards IEEE 802.11. In addition, although Figure 1b depicted radio 102c as operating within SE 102, a device 101 can also support a radio 102c operating outside SE 102 and connected to CPU 101c via bus 101g.

Figure 6b

Figure 6b is a graphical illustration of components for a device, in accordance with exemplary embodiments. Device 101 as depicted in the above Figure 1a through Figure 6a may comprise several different physical and logical embodiments in the present invention. Figure 6b depicts a different physical configuration for device 101 than that depicted in Figure 1b and 6a, while device 101 can still utilize the components and sub-steps for a configuration step 103 as described herein. As depicted in Figure 6b, a device 101 could include a “system on a chip” (SOC) 601, a transducer 105, and a Radio RF 602b. SOC 601 could comprise a package soldered onto a circuit board within device 101 which integrates the various components depicted for SOC 601. Radio RF 602b can comprise a radio front end for a radio 101z, where radio RF 602b is coupled to radio baseband 602a, where radio baseband 602a comprises a baseband processor for a radio RF 602b. In exemplary

embodiments, radio RF 602b and radio baseband 602a can operate in conjunction to provide the functionality of a radio 101z as depicted and described in connection with Figure 1b. The transducer 105 in a device 101 could comprise a transducer 105 as depicted and described in connection with Figure 1b above, including any of transducer 105 that
5 comprises a bi-directional transducer 105a, transducer input 105b, or transducer output 105c.

A system on a chip (SOC) 601 in Figure 6b could include a processor CPU 101c, a secure processing environment 102, a transducer interface 102h, nonvolatile memory storage 101f, random access memory 101d, and radio baseband 602a, where the
10 components for SOC 601 can be linked via bus 601a. Bus 601a could comprise a data bus similar to bus 101g or bus 102i as depicted and described in connection with Figure 1b above, and bus 601a can have bandwidth and signal connections necessary to support the internal operation of the different components within SOC 601. The processor CPU 101c, SE 102, transducer interface 102h, and random access memory 101d can operate and
15 provide the functionality as depicted and described in connection with Figure 1b above for the same elements shown. Although CPU 101c and SE 102 are depicted as separate elements in Figure 6b, in exemplary embodiments CPU 101c and SE 102 within a SOC 601 could be combined into a single processing environment, such as the combined CPU 101c and SE 102 as depicted and described in connection with Figure 6a. Either (X) CPU
20 101c and SE 102 could operate with separate physical transistors, logic gates, and memory registers in exemplary embodiments, or (Y) SE 102 and CPU 101c could share physical resources within SOC 601 and operate in a logically separate manner. In other exemplary embodiments for a device 101 in Figure 6a and also other figures herein, a separate SE 102 could be optionally omitted, and the functionality of SE 102 could be provided by a CPU
25 101c. As one example, instead of a set of cryptographic algorithms 141 within SE 102 as depicted in Figure 1b being loaded into an EEPROM 102j, the set of cryptographic algorithms 141 could be performed by CPU 101c and stored in a nonvolatile memory 101f for device 101. Other possibilities exist as well for the configuration of a CPU 101c and SE 102 without departing from the scope of the present invention.

30 In exemplary embodiments such as that depicted in Figure 6b, the function of an NFC radio 102c could operate within separate elements within a device 101. The operation of a radio 102c could be divided into a RF baseband 102ca processor and an RF front end 102cb. RF baseband 102ca can provide baseband processing functionality for radio 102c, and RF front end 102cb can provide RF front end functionality such as RF filters and RF
35 amplifiers. RF front end 102cb can be connected to an antenna, which could also be located inside the physical housing of device 101, or an antenna can be external to device

101 and electrically connected to radio RF 602b. The divided NFC radio 102c could also support other relatively short-range wireless communication protocols appropriate for a device 101 communicating with a mobile handset 108. Other protocols besides NFC could comprise Bluetooth, Bluetooth Low Energy (BLE), ZigBee, Z-Wave, 6LoWPAN, Thread, 5 Wi-Fi, Wi-Fi-ah (HaLow), LoRA, and ANT (from ANT Wireless). Other possibilities exist as well for a short range wireless communications protocol supported by radio 102c (or RF baseband 102ca and RF front end 102cb) without departing from the scope of the present invention.

10 **Conclusion**

Various exemplary embodiments have been described above. Those skilled in the art will understand, however, that changes and modifications may be made to those examples without departing from the scope of the claims.

CLAIMS

1. A method for a device to securely receive configuration data, the method performed by the device, the method comprising:
- a) recording in a protected memory, a first device private key for an elliptic curve Diffie-Hellman (ECDH) key exchange algorithm, a second device private key for a digital signature algorithm, a certificate authority public key, and a first set of cryptographic parameters, wherein the first device private key corresponds to a first device public key, and wherein the second device private key corresponds to a second device public key;
 - b) establishing a wireless connection, wherein the wireless connection is encrypted;
 - c) receiving via the wireless connection, (i) a first server public key for a server (ii) a second set of cryptographic parameters, and (iii) a random number;
 - d) authenticating (i) the server using at least the certificate authority public key, and (ii) the device using at least the received random number and the second device private key;
 - e) deriving a first symmetric ciphering key using at least (i) the first device private key and the first server public key, (ii) the ECDH key exchange algorithm, and (iii) the first set of cryptographic parameters;
 - f) deriving a third device private key and a third device public key using the second set of cryptographic parameters;
 - g) generating a device digital signature over at least the third device public key using the second device private key;
 - h) encrypting at least the derived third device public key into a first ciphertext using the derived first symmetric ciphering key;

- i) sending, via the wireless connection, the first ciphertext and the device digital signature to the server; and
- j) receiving, via the wireless connection, a second ciphertext of a first device configuration, wherein the device decrypts the second ciphertext using at least the first symmetric ciphering key.

2. The method of claim 1, wherein the device includes a secure element, and wherein the secure element for the device conducts steps a), d), e), f), g), and h).

3. The method of claim 2, wherein the protected memory comprises a nonvolatile memory within the secure element with a memory address, and wherein a processor for the device is prevented from reading from the memory address.

4. The method of any one of claims 1 to 3, wherein the first device configuration includes (i) an updated configuration for the secure element and (ii) a device certificate for the derived third device public key.

5. The method of claim 4, wherein a boot firmware for the secure element uses the first device private key for the ECDH key exchange algorithm and the second device private key for the digital signature algorithm, wherein the secure element uses the updated configuration with the device certificate.

6. The method of any one of claims 1 to 5, wherein, after a step j), the device authenticates by using the digital signature algorithm with the first device configuration and the derived third device private key.

7. The method of any one of claims 2, 3, 5, and 6, wherein the first device configuration includes a device certificate for the derived third device public key.

8. The method of claim 7, further comprising: verifying, via the device, the device certificate using at least the certificate authority public key.

9. The method of claim 7 or 8, further comprising: storing, via the device, the device certificate in the protected memory, and using, via the device, the device certificate and the derived third device private key in order to authenticate with at least one of (i) an access network, (ii) a reporting system, and (iii) an authentication server.

10. The method of any one of claims 1 to 9, further comprising:

k) receiving via the wireless connection, a second server public key for the second set of cryptographic parameters;

l) deriving a second symmetric ciphering key using (i) the derived third device private key the second server public key, (ii) the ECDH key exchange algorithm, and (iii) the second set of cryptographic parameters; and

m) receiving a third ciphertext of a second device configuration, wherein the second ciphertext is decrypted by the device using the derived second symmetric ciphering key.

11. The method of claim 10, wherein the second device configuration includes a set of network access credentials, and wherein the device authenticates with a wireless access network using the set of network access credentials.

12. The method of claim 10 or 11, wherein the second device configuration includes at least one file for the device, wherein the at least one file is used by the device with at least one of a device operating system, a device reporting application, a secure element firmware, a secure element operating system, a transducer library, and a configuration test vector.

13. The method of claim 12, wherein a secure element boot firmware uses the first device private key for the ECDH key exchange algorithm and the second device private key for the digital signature algorithm, and wherein the device uses the second device configuration with the secure element firmware.

14. The method of any one of claims 2 to 5, wherein the secure element sends and receives through the device using an external bus controller for the secure element, wherein the device and the secure element are connected via a data bus, and wherein the device sends and receives through a radio.

15. The method of any one of claims 1 to 14, wherein the first set of cryptographic parameters specifies at least a first elliptic curve defining equation, and wherein the second set of cryptographic parameters specifies at least a second elliptic curve defining equation.

16. The method of any one of claims 1 to 15, wherein the device establishes the wireless connection with one of (i) a mobile handset through a local wireless connection and (ii) the server through a wide area wireless connection.

17. The method of claim 16, wherein the local wireless connection comprises one of (i) a Wi-Fi connection, (ii) a Bluetooth connection, and (iii) a Near Field Communications (NFC) connection.

18. The method of claim 16, wherein the wide area wireless connection comprises one of (i) a 4G network connection, and (ii) a 5G network connection.

19. The method of any one of claims 1 to 18, wherein the device includes a tag, wherein the tag specifies a uniform resource locator (URL), wherein a configuration system uses the URL in order to receive (i) a device identity from the tag, and (ii) at least one of the first device public key and the second device public key, and wherein the configuration system includes at least the server.

20. The method of claim 19, wherein a mobile handset reads the tag using at least one of a camera and a NFC reader and sends a tag value from the tag to the configuration system.

21. The method of claim 19 or claim 20, wherein the configuration system uses the device identity to query a database for at least the first device public key and the second device public key.

22. The method of any one of claims 1 to 21, further comprising: conducting, via the device, step a) before distributing the device to a user.

23. The method of any one of claims 1 to 22, wherein the device generates the device digital signature over at least the third device public key and the random number using the second device private key.

24. The method of any one of claims 1 to 23, wherein the server derives the first symmetric ciphering key using (i) a first server private key for the first server public key and the first device public key, (ii) the ECDH key exchange algorithm, and (iii) the first set of cryptographic parameters, and wherein the server uses the first symmetric ciphering key to decrypt the first ciphertext and encrypt the second ciphertext.

25. The method of any one of claims 2 to 5, wherein the secure element includes a hardware random number generator, and wherein the hardware random number generator uses at least a transducer measurement in order to generate a device random number, and wherein the secure element uses the device random number to derive the third private key.

26. The method of any one of claims 1 to 25, wherein the device does not send the first and second device public keys.

27. The method of any one of claims 1 to 26, wherein the digital signature algorithm comprises an elliptic curve digital signature algorithm (ECDSA).

28. The method of any one of claims 1 to 27, wherein the device receives via the wireless connection, (i) the first server public key for the server (ii) the second set of cryptographic parameters, and (iii) the random number with a server digital signature, and wherein the device authenticates the server digital signature using at least the certificate authority public key.

29. The method of any one of claims 1 to 28, wherein the first device configuration includes least one of a certificate authority certificate, a transducer configuration, a monitored unit configuration, an identity list, a device configuration, and a reporting system configuration.

30. The method of claim 29, wherein the identity list includes at least one of a networks-available list, a transducer identity, a device owner identity, a device location value, and a monitored unit identity.

31. The method of claim 9, wherein the access network comprises a 5G network supporting an EAP-TLS authentication, and the method further comprising: conducting, via the device, a connection setup with the device certificate and the third device private key.

32. The method of any one of claims 1 to 31, wherein the first device configuration includes a set of network access credentials, and wherein the device authenticates with a wireless access network using the set of network access credentials.

33. The method of any one of claims 1 to 32, wherein the device authenticates the device (i) using the received random number and the second device private key, and (ii) sending the second device public key with an authenticating digital signature.

34. The method of any one of claims 1 to 32, wherein the device authenticates the device (i) using the received random number and the second device private key, and (ii) sending an authenticating digital signature.

35. The method of any one of claims 1 to 34, wherein the device does not use any of the first device private key, the second device private key, and the certificate authority public key in order to establish the encrypted wireless connection.

36. The method of any one of claims 1 to 35, wherein the first set of cryptographic parameters and the second set of cryptographic parameters are the same.

37. The method of any one of claims 1 to 36, wherein the device receives at least one of (i) the second set of cryptographic parameters and (ii) the first server public key, after the device authenticates the device.

38. The method of any one of claims 1 to 37, wherein the first server public key comprises an ephemeral key.

39. The method of any one of claims 1 to 38, wherein the device receives a server certificate in step c), and wherein the device authenticates the server using at least the certificate authority public key by (i) verifying a server digital signature over at least one of the first server public key and the random number using the server certificate, and (ii) verifying the server certificate using the certificate authority public key.

40. The method of any one of claims 1 to 39, wherein the device derives the first device private key after establishing the wireless connection, and wherein the device records the first device private key after deriving the first device private key.

41. The method of any one of claims 1 to 40, wherein the second set of cryptographic parameters specifies values for algorithms associated with a key encapsulation mechanism, and wherein the algorithms comprise one of lattice-based cryptography, code-based cryptography, and supersingular elliptic curve isogeny cryptography.

42. The method of any one of claims 1 to 41, wherein the third device public key supports at least one of lattice-based cryptography, code-based cryptography, supersingular elliptic curve isogeny cryptography, and elliptic curve cryptography.

43. The method of claim 4 or 5, wherein the updated configuration includes at least one file for the secure element, wherein the at least one file is used by the secure element with at least one of a secure element firmware, a secure element operating system, a transducer library, and a configuration test vector.

44. The method of any one of claims 1 to 43, wherein the device receives via the wireless connection, (i) the first server public key for the server (ii) the second set of cryptographic parameters, and (iii) the random number and an authentication token with a server digital signature, wherein the device authenticates the server digital signature using at least the certificate authority public key, wherein the device records the authentication token in a step a), and wherein the device does not send the authentication token via the wireless connection.

45. The method of claim 44, wherein the configuration system receives an identity token for the device before conducting step c), and wherein the configuration system uses the identity token to query a database for the authentication token, and wherein the configuration system generates the server digital signature over at least the authentication token.

46. The method of claim 45, wherein the authentication token comprises at least one of a device identity and an initial random number.

47. The method of any one of claims 1 to 46, wherein the wireless connection is encrypted before the device receives the first server public key and the second set of cryptographic parameters.

48. The method of any one of claims 1 to 47, wherein the device derives the third public key before the device authenticates the server.

49. The method of claim 48, wherein the device generates the device digital signature before the device authenticates the server.

50. The method of any one of claims 1 to 49, wherein the device generates the device digital signature over at least the third device public key using the second device private key and the first set of cryptographic parameters.

51. The method of any one of claims 1 to 49, wherein the device generates the device digital signature over at least the third device public key using the second device private key and the second set of cryptographic parameters.

52. A method for a device to securely receive configuration data, the method performed by the device, the method comprising:

- a) recording in a protected memory, a first device private key for a key exchange algorithm, a second device private key for a digital signature algorithm, a certificate authority public key, and a first set of cryptographic parameters, wherein the first device private key corresponds to a first device public key, and wherein the second device private key corresponds to a second device public key;
- b) establishing a wireless connection, wherein the wireless connection is encrypted;
- c) receiving via the wireless connection, (i) a first server public key for a server (ii) a second set of cryptographic parameters, and (iii) a random number;
- d) authenticating (i) the server using at least the certificate authority public key, and (ii) the device using at least the received random number and the second device private key;
- e) deriving a first symmetric ciphering key using at least (i) the first device private key and the first server public key, (ii) the key exchange algorithm, and (iii) the first set of cryptographic parameters;

- f) deriving a third device private key and a third device public key using the second set of cryptographic parameters;
- g) generating a device digital signature over at least the third device public key using the second device private key;
- h) encrypting at least the derived third device public key into a first ciphertext using the derived first symmetric ciphering key;
- i) sending, via the wireless connection, the first ciphertext and the device digital signature to the server; and
- j) receiving, via the wireless connection, a second ciphertext of a first device configuration, wherein the device decrypts the second ciphertext using at least the first symmetric ciphering key.

53. The method of claim 52 and any one of claims 2 to 4.

54. The method of claim 52, wherein a boot firmware for the secure element uses the first device private key for the key exchange algorithm and the second device private key for the digital signature algorithm, wherein the secure element uses the updated configuration with the device certificate.

55. The method of claim 52, and any one of claims 53 and 54.

56. The method of any one of claims 52 to 54, further comprising:

- k) receiving via the wireless connection, a second server public key for the second set of cryptographic parameters;
- l) deriving a second symmetric ciphering key using (i) the derived third device private key the second server public key, (ii) the key exchange algorithm, and (iii) the second set of cryptographic parameters; and,

m) receiving a third ciphertext of a second device configuration, wherein the second ciphertext is decrypted by the device using the derived second symmetric ciphering key.

57. The method of claim 52, and any one of claims 11 to 23.

58. The method of any one of claims 52 to 57, wherein the server derives the first symmetric ciphering key using (i) a first server private key and the first device public key, (ii) the key exchange algorithm, and (iii) the first set of cryptographic parameters, and wherein the server uses the first symmetric ciphering key to decrypt the first ciphertext and encrypt the second ciphertext;

59. The method of claim 52, and any one of claims 25 to 51.

60. A method for a configuration system to securely send configuration data to a device, the method performed by the configuration system, the method comprising:

- a) storing a first server private key for an elliptic curve Diffie-Hellman (ECDH) key exchange algorithm, a second server private key for a digital signature algorithm, a first set of cryptographic parameters, wherein the first server private key corresponds to a first server public key, and wherein the second server private key corresponds to a second server public key;
- b) receiving an identity token for a device, a first device public key for the ECDH key exchange algorithm, and a second device public key for the digital signature algorithm;
- c) sending to the device (i) the first server public key for a server (ii) a second set of cryptographic parameters, and (iii) a random number;
- d) authenticating (i) the server with the device using at least the second server private key, and (ii) the device with the server using at least the second device public key;

- e) deriving a first symmetric ciphering key using (i) the first device public key and the first server private key, (ii) the ECDH key exchange algorithm, and (iii) the first set of cryptographic parameters;
- f) receiving, from the device, (i) a first ciphertext of a third device public key and (ii) a device digital signature for at least the third device public key, wherein the third device public key supports the second set of cryptographic parameters;
- g) decrypting the first ciphertext using the derived first symmetric ciphering key in order to read the plaintext third device public key;
- h) verifying the device digital signature using at least the second device public key;
- i) selecting first configuration data for the device using at least the identity token;
- j) encrypting a second ciphertext of at least the first configuration data using the derived first symmetric ciphering key; and
- k) sending, to the device, the second ciphertext.

61. The method of claim 60, further comprising:

- l) generating, before encrypting the second ciphertext, a device certificate for the third device public key, wherein the device certificate includes a certificate authority digital signature, and wherein the first configuration data includes the device certificate.

62. The method of claim 61, wherein, after a step l), the configuration system authenticates the device using at least the device certificate.

63. The method of claim 60, further comprising:

- l) deriving a third server private key and a third server public key for the second set of cryptographic parameters;
- m) sending, to the device, the third server public key;
- n) deriving a second symmetric ciphering key using (i) the derived third server private key and the third device public key, (ii) the ECDH key exchange algorithm, and (iii) the second set of cryptographic parameters;
- o) encrypting a third ciphertext of second configuration data, wherein the second ciphertext is encrypted using the derived second symmetric ciphering key; and
- p) sending, to the device, the third ciphertext.

64. The method of claim 63, further comprising:

- q) generating, before encrypting the third ciphertext, a device certificate for the third device public key, wherein the device certificate includes a certificate authority digital signature, and wherein the second configuration data includes the device certificate.

65. The method of claim 63 or 64, wherein the second configuration data includes a set of network access credentials, and wherein, after a step p), an access network authenticates the device using the network access credentials.

66. The method of any one of claims 63 to 65, wherein the second configuration data includes at least one file for the device, wherein the at least one file is used by the device with at least one of a device operating system, a device reporting application, a secure element firmware, a secure element operating system, a transducer library, and a configuration test vector.

67. The method of any one of claims 60 to 66, wherein the first set of cryptographic parameters specifies at least a first elliptic curve defining equation, and wherein the second set of cryptographic parameters specifies at least a second elliptic curve defining equation.

68. The method of any one of claims 60 to 67, wherein the device includes a tag, wherein the tag specifies a uniform resource locator (URL), wherein the configuration system uses the URL in order to receive the identity token for the device.

69. The method of claim 68, wherein a mobile handset reads the tag using at least one of a camera and a NFC reader and the configuration system receives from the mobile handset a tag value from the tag.

70. The method of any one of claims 60 to 69, wherein the configuration system receives the identity token from the device.

71. The method of claim 70, wherein the configuration system uses the identity token to query a database for the first device public key and the second device public key, wherein the configuration system receives the first device public key and the second device public key from the database, and wherein the device does not send first device public key and the second device public key.

72. The method of claim 71, wherein the configuration system includes the database, and wherein the database stores the first device public key and the second device public key before the configuration system receives the identity token for the device.

73. The method of claim 70, wherein the configuration system receives from the device at least one of the first device public key, the second device public key, and the first set of cryptographic parameters.

74. The method of any one of claims 60 to 73, wherein the identity token for the device comprises one of (i) a device identity, or (ii) an obfuscated device identity.

75. The method of any one of claims 60 to 74, wherein the configuration system, after receiving the identity token, (i) derives the first server public key and first server private key, and (ii) stores the first server public key and first server private key, wherein the first server private key comprises an ephemeral key, and wherein the first server public key comprises an ephemeral key.

76. The method of any one of claims 60 to 75, wherein the configuration system includes (1) a discovery server for at least receiving the identity token for the device, an authentication server for authenticating (i) the server with the device and (ii) the device with the server, and (2) the server for at least (i) deriving the first symmetric ciphering key, (ii) receiving the first ciphertext, and (iii) sending the second ciphertext.

77. The method of any one of claims 60 to 76, wherein authenticating the server with the device comprises (1) using the second server private key and the digital signature algorithm to generate a server digital signature over at least one of (i) the first server public key, (ii) identity token, and (iii) the random number, and (2) sending to the device the server digital signature.

78. The method of any one of claims 60 to 77, wherein, before deriving the first symmetric ciphering key, the configuration system sends to the device the second server public key in a server certificate, and wherein the device verifies the server certificate with a certificate authority public key recorded in the device.

79. The method of any one of claims 60 to 77, wherein, before the device communicates with the configuration system, the device records the second server public key.

80. The method of any one of claims 60 to 79, wherein authenticating the device with the server comprises (i) receiving from the device an authentication digital signature over at least the random number, and (ii) verifying the authentication digital signature using at least the second device public key.

81. The method of any one of claims 60 to 80, wherein the device derives the third device public key after receiving the second set of cryptographic parameters.

82. The method of any one of claims 60 to 81, wherein the configuration system receives from the device the device digital signature over at least the third device public key and the random number.

83. The method of any one of claims 60 to 82, wherein the first ciphertext includes the device digital signature.

84. The method of any one of claims 60 to 83, wherein the device derives the first symmetric ciphering key using (i) the first server public key and a first device private key, (ii) the ECDH key exchange algorithm, and (iii) the first set of cryptographic parameters, and wherein the device uses the first symmetric ciphering key to encrypt the first ciphertext and decrypt the second ciphertext.

85. The method of any one of claims 60 to 84, wherein the digital signature algorithm comprises an elliptic curve digital signature algorithm (ECDSA).

86. The method of any one of claims 60 to 85, wherein the configuration system queries at least one database using the identity token in order to select the first configuration data.

87. The method of any one of claims 60 to 86, wherein the first configuration data includes least one of a certificate authority certificate, a transducer configuration, a monitored unit configuration, an identity list, a device configuration, and a reporting system configuration.

88. The method of claim 87, wherein the identity list includes at least one of a networks available list, a transducer identity, a device owner identity, a device location value, and a monitored unit identity.

89. The method of claim 61 or 64, wherein an access network associated with the configuration system supports an EAP-TLS authentication, and wherein the access network conducts a connection setup with the device using the device certificate for the third device public key.

90. The method of claim 89, wherein the access network comprises a 5G wireless network.

91. The method of any one of claims 60 to 88, wherein the first configuration data includes a set of network access credentials, and wherein an access network associated with the configuration system authenticates the device using the network access credentials.

92. The method of any one of claims 60 to 91, wherein the first set of cryptographic parameters and the second set of cryptographic parameters are the same.

93. The method of any one of claims 60 to 92, wherein the configuration system sends to the device at least one of (i) the second set of cryptographic parameters and (ii) the first server public key after the configuration system authenticates the device with the server.

94. The method of claim 60, further comprising:

- l) deriving a third server private key and a third server public key for the first set of cryptographic parameters;
 - m) sending, to the device, the third server public key;
 - n) deriving a second symmetric ciphering key using (i) the derived third server private key and the first device public key, (ii) the ECDH key exchange algorithm, and (iii) the first set of cryptographic parameters;
 - o) encrypting a third ciphertext of second configuration data, wherein the second ciphertext is encrypted using the derived second symmetric ciphering key;
- and

p) sending, to the device, the third ciphertext.

95. The method of claim 94, wherein the second configuration data includes a set of network access credentials, and wherein, after a step p), an access network authenticates the device using the set of network access credentials.

96. The method of claim 94 or 95, wherein the second configuration data includes at least one file for the device, wherein the at least one file is used by the device with at least one of a device operating system, a device reporting application, a secure element firmware.

97. The method of claim 94 and any one of claims 64 to 93.

98. The method of any one of claims 60 to 97, wherein the configuration system sends the third device public key to an access network, wherein the access network (i) encrypts a second symmetric ciphering key with the third device public key, (ii) encrypts a set of network access credentials with the second symmetric ciphering key, and (iii) the configuration system receives the encrypted set of network access credentials, wherein the configuration system includes the encrypted set of network access credentials in the second ciphertext, wherein the device decrypts the encrypted set of network access credentials with a private key corresponding to the third device public key, and wherein the access network authenticates the device using the set of network access credentials.

99. The method of any one of claims 60 to 98, wherein the configuration system sends the first server public key after authenticating the device with the server.

100. The method of any one of claims 60 to 99, wherein the configuration system receives the second device public key before storing the first server private key.

101. The method of any one of claims 60 to 100, wherein the configuration system selects the first configuration data before sending to the device the first server public key.

102. The method of any one of claims 60 to 101, wherein the configuration system receives the random number from the device, and wherein the configuration system receives the random number with the identity token.

103. The method of any one of claims 60 to 101, wherein the configuration system generates the random number.

104. The method of any one of claims 60 to 103, wherein the configuration system derives the first server private key and stores the first server private key before the configuration system authenticates the server with the device.

105. The method of any one of claims 60 to 104, wherein the configuration system derives the first server private key and stores the first server private key after the configuration system authenticates the server with the device.

106. The method of any one of claims 60 to 105, wherein the configuration system receives at least one of the first device public key and the second device public key after sending to the device at least one of (i) the first server public key, (ii) the second set of cryptographic parameters, and (iii) the random number.

107. A method for a mobile handset to configure a device, the method performed by the mobile handset, the method comprising:

- a) reading a tag value for the device, wherein the tag value includes an identity token for the device;
- b) establishing a secure session with a configuration system using at least the tag value;
- c) sending the identity token to the configuration system;
- c) establishing a local wireless connection with the device;

- d) receiving from the configuration system and sending to the device: a first random number, a first certificate for an authentication server, and a first digital signature over at least the first random number and an authentication token;
- e) receiving from the device and sending to the configuration system: a second random number and a second digital signature over at least the first random number;
- f) receiving from the configuration system and sending to the device a first public key of the configuration system, a set of cryptographic parameters, a second certificate for the configuration system, and a third digital signature over at least the second random number;
- g) receiving from the device and sending to the configuration system, a first ciphertext, wherein the first ciphertext is encrypted using at least the first public key; and
- h) receiving from the configuration system and sending to the device, a second ciphertext of a device configuration for the device.

108. The method of claim 107, wherein the mobile handset uses at least one of a camera and a NFC reader to read the tag value, and wherein the tag value includes at least a uniform resource locator (URL) for the device, and wherein the mobile handset establishes the secure session with the configuration system using at least the URL.

109. The method of claim 107, wherein the mobile handset reads the tag value from input by a user of the mobile handset.

110. The method of any one of claims 107 to 109, wherein the local wireless connection comprises one of (i) a Wi-Fi connection, (ii) a Bluetooth connection, and (iii) a Near Field Communications (NFC) connection.

111. The method of claim 107, wherein the mobile handset establishes the local wireless connection before reading the tag value, and wherein the mobile handset receives the tag value through the local wireless connection, and wherein the mobile handset reads the tag value from a data packet received through the local wireless connection.

112. The method of claim 107, wherein the mobile handset establishes the local wireless connection before the mobile handset establishes the secure session.

113. The method of any one of claims 107 to 112, wherein the mobile handset establishes the secure session over one of (i) a wide area wireless connection, and where the wide area wireless connection comprises at least one of a 4G network connection and a 5G network connection, and (ii) a Wi-Fi connection with connectivity to the Internet.

114. The method of claim 113, wherein the secure session uses at least one of (i) transport layer security (TLS), datagram transport layer security (DTLS), Internet Protocol Security (IPSec), a virtual private network (VPN), and a secure shell (SSH).

115. The method of any one of claims 107 to 114, wherein a tag includes the tag value, and wherein the tag includes the identity token for the device, and wherein the mobile handset reads the tag.

116. The method of any one of claims 107 to 115, wherein the mobile handset sends the identity token to the configuration system through the secure session.

117. The method of any one of claims 107 to 116, wherein the mobile handset receives the first random number from the device and sends the first random number to the configuration system, before the mobile handset receives the first digital signature from the configuration system.

118. The method of any one of claims 107 to 117, wherein the device verifies the first digital signature using at least the first certificate, and wherein the device records the authentication token before the local wireless connection is established.

119. The method of any one of claims 107 to 118, wherein the mobile handset does not receive the first random number from the device before receiving the second digital signature from the device, wherein the device verifies the first digital signature using at least the first certificate before the mobile handset receives the second digital signature, and wherein the device records the authentication token.

120. The method of any one of claims 107 to 119, wherein the configuration system uses the identity token for the device to query a database for a first device public key and the authentication token, and wherein the configuration system uses the first device public key to verify the second digital signature.

121. The method of claim 120, wherein the first device public key is recorded in a first device certificate.

122. The method of claim 120, wherein the mobile handset does not receive the first device public key from the device.

123. The method of any one of claims 1 to 118, wherein the mobile handset receives a first device public key from the device via the local wireless connection, and wherein the mobile handset sends the first device public key to the configuration system before receiving the second ciphertext.

124. The method of claim 123, wherein the configuration system uses the first device public key to verify the second digital signature.

125. The method of any one of claims 107 to 124, wherein the identity token for the device and the authentication token are different, and wherein the authentication token is recorded by the device before step b), and wherein the configuration system uses the identity token to query a database for the device value.

126. The method of any one of claims 107 to 124, wherein the identity token for the device and the authentication token are the same.

127. The method of any one of claims 107 to 126, wherein third digital signature is over the second random number and at least one of the first public key and the set of cryptographic parameters.

128. The method of any one of claims 107 to 127, wherein the device verifies the third digital signature using at least the second certificate for the configuration system.

129. The method of any one of claims 107 to 128, wherein the first ciphertext includes a second public key derived by the device, and wherein the device configuration in the second ciphertext includes a device certificate for the second public key.

130. The method of claim 120 or 123, wherein the mobile handset receives from the device the first ciphertext with a fourth digital signature, and wherein the configuration system verifies the fourth digital signature using at least the first device public key.

131. The method of claim 130, wherein the first ciphertext includes a second public key derived by the device, and wherein the device configuration in the second ciphertext includes a device certificate for the second public key.

132. The method of claim 131, wherein the device authenticates with a reporting system using the device certificate.

133. The method of claim 131, wherein the device authenticates with an access network using the device certificate

134. The method of any one of claims 107 to 133, wherein the device mutually derives a symmetric ciphering key with the configuration system using the first public key and a device private key, and wherein the configuration system mutually derives the symmetric ciphering key and decrypts the first ciphertext using the symmetric ciphering key.

135. The method of claim 134, wherein the device encrypts the first ciphertext using the symmetric ciphering key and the configuration system decrypts the first ciphertext using the symmetric ciphering key, and wherein the configuration system encrypts the second ciphertext using the symmetric ciphering key and the device decrypts the second ciphertext using the symmetric ciphering key.

136. The method of claim 134 or 135, wherein the device mutually derives the symmetric ciphering key using an elliptic curve Diffie-Hellman (ECDH) key exchange and the set of cryptographic parameters.

137. The method of any one of claims 107 to 136, wherein the device configuration includes a set of network access credentials, and wherein, after a step h), an access network authenticates the device using the network access credentials.

138. The method of any one of claims 107 to 137, wherein the device configuration includes at least one file for the device, wherein the file is used by the device with at least one of a device operating system, a device reporting application, a secure element firmware, a secure element operating system, a transducer library, and a configuration test vector

139. The method of any one of claims 107 to 138, wherein the device configuration includes least one of a certificate authority certificate, a transducer configuration, a monitored unit configuration, an identity list, a device configuration, and a reporting system configuration.

140. The method of claim 139, wherein the identity list includes at least one of a networks available list, a transducer identity, a device owner identity, a device location value, and a monitored unit identity.

141. The method of any one of claims 107 to 140, wherein the first ciphertext includes an initial device configuration, wherein the device updates the initial device configuration using the device configuration in the second ciphertext.

142. The method of claim 141, wherein the configuration system uses the initial device configuration in order to select the device configuration.

143. The method of any one of claim 107 to 142, wherein the device records the authentication token before a step b).

Figure 1a

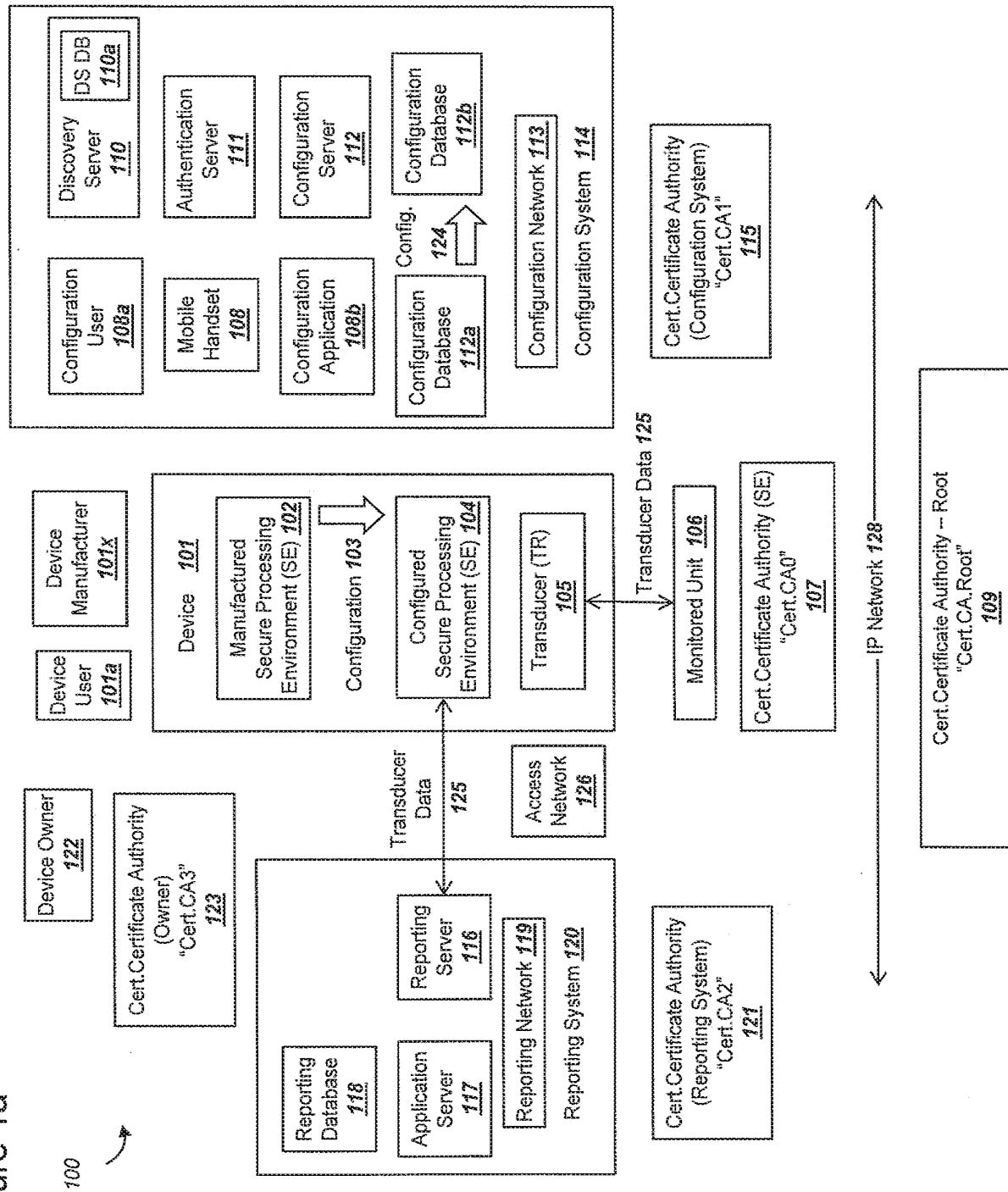
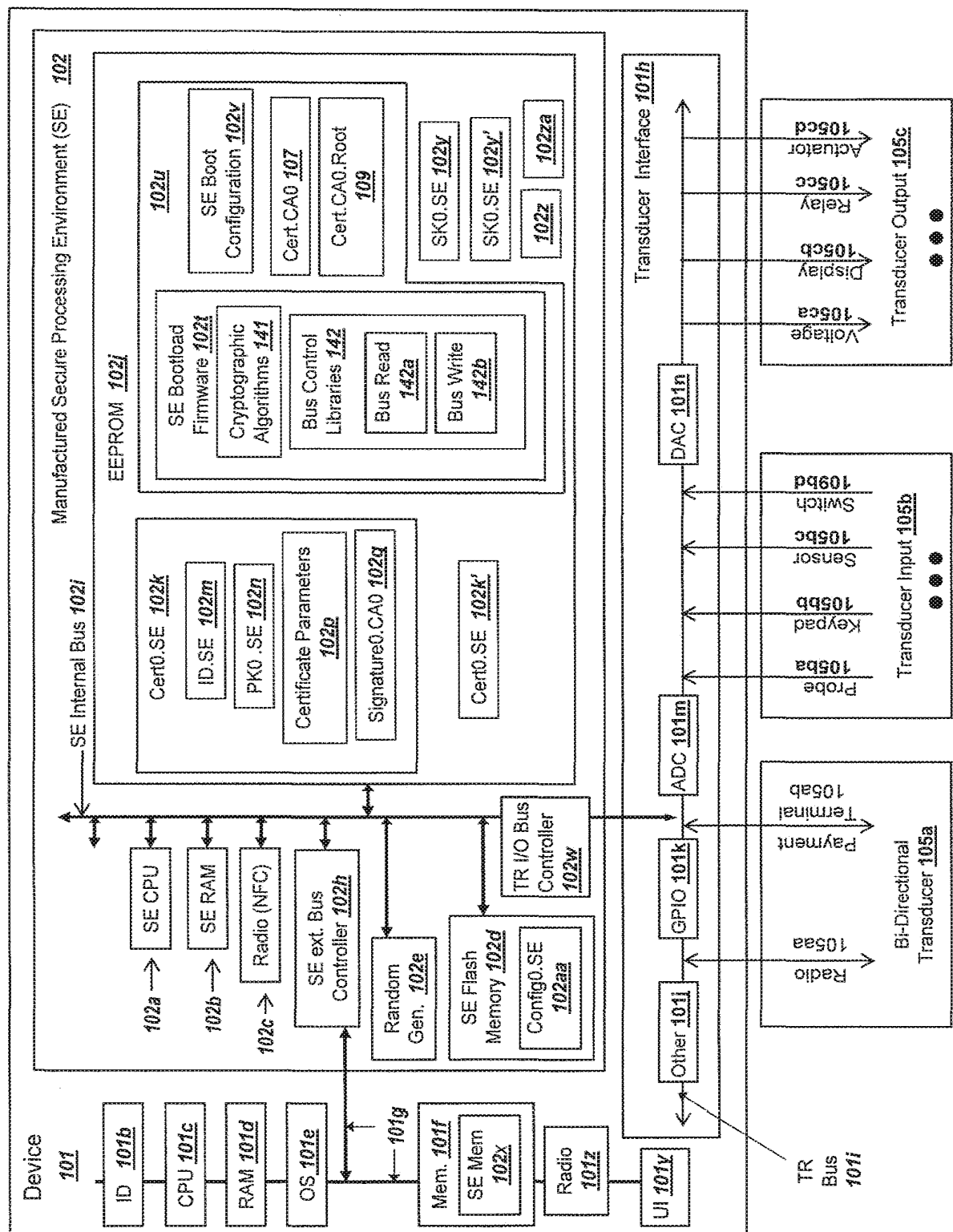


Figure 1b



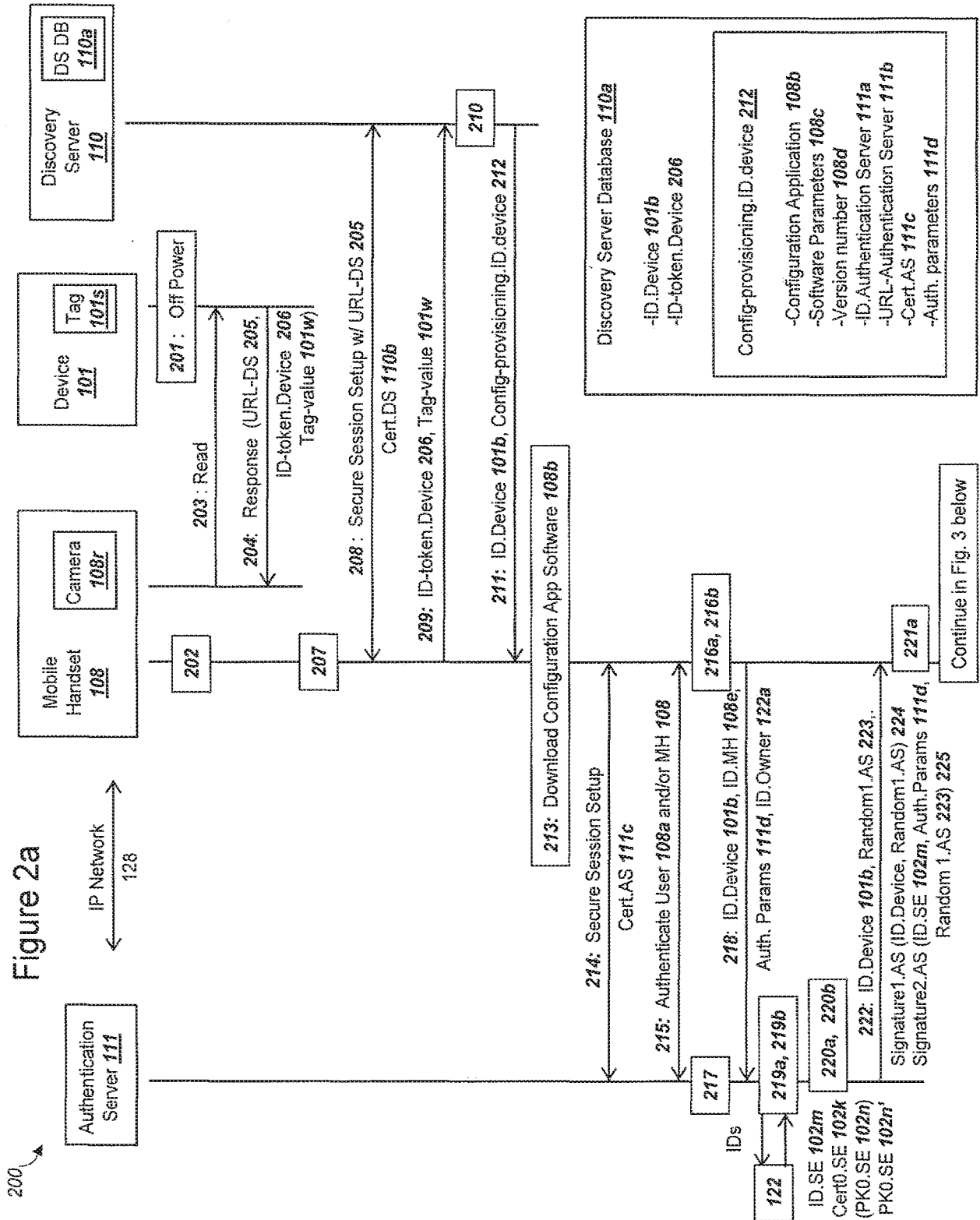


Figure 2b

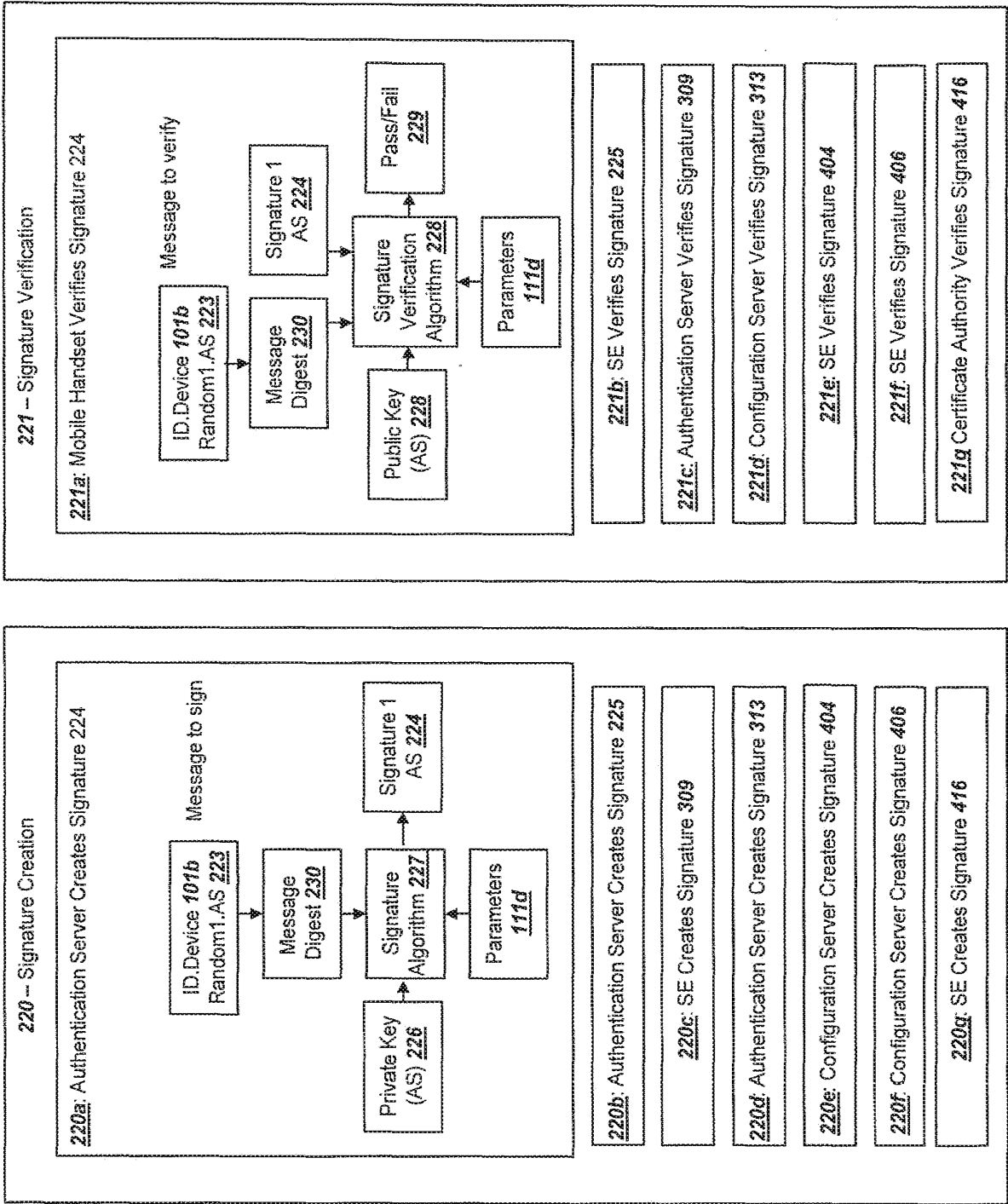


Figure 3

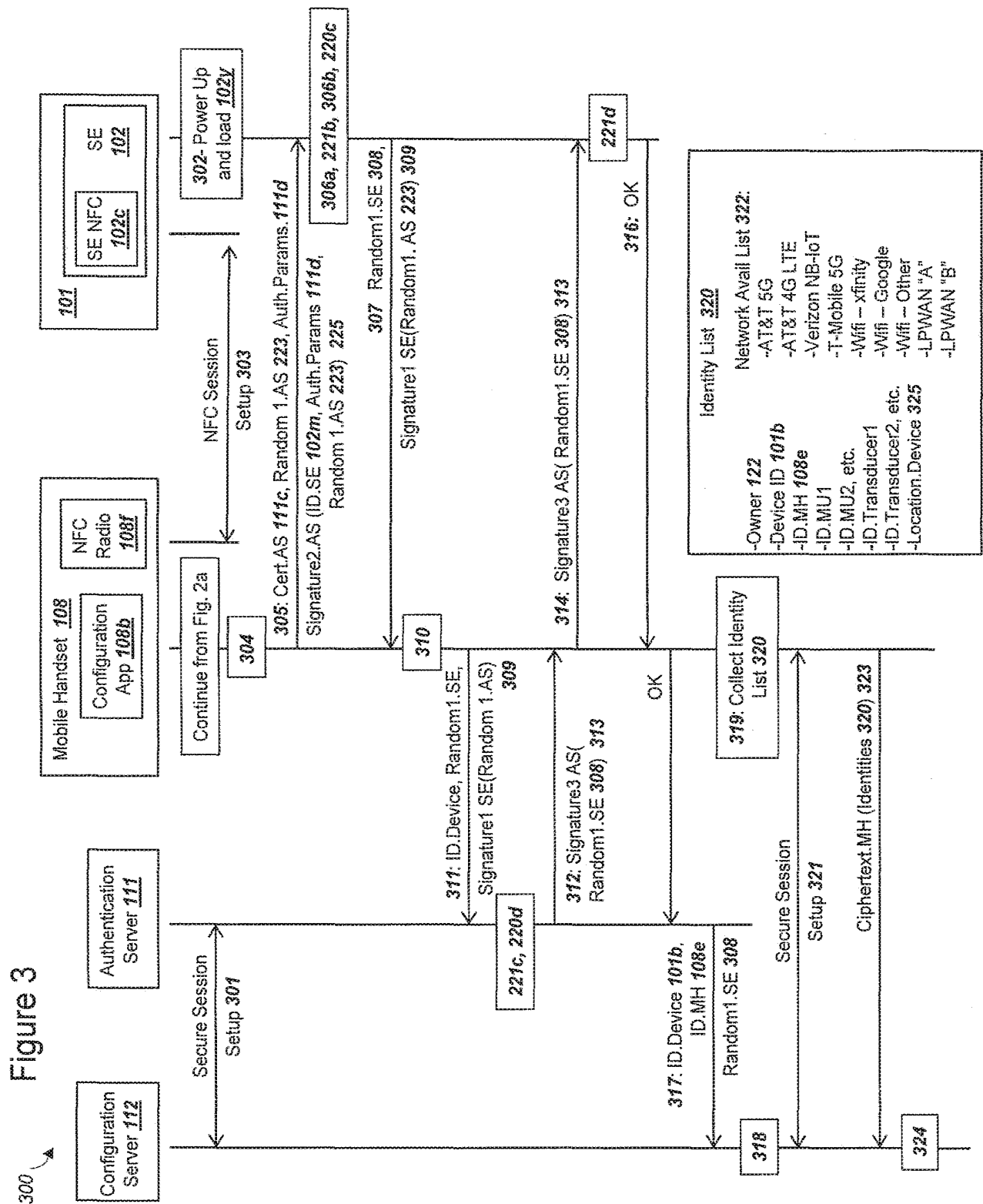


Figure 4b

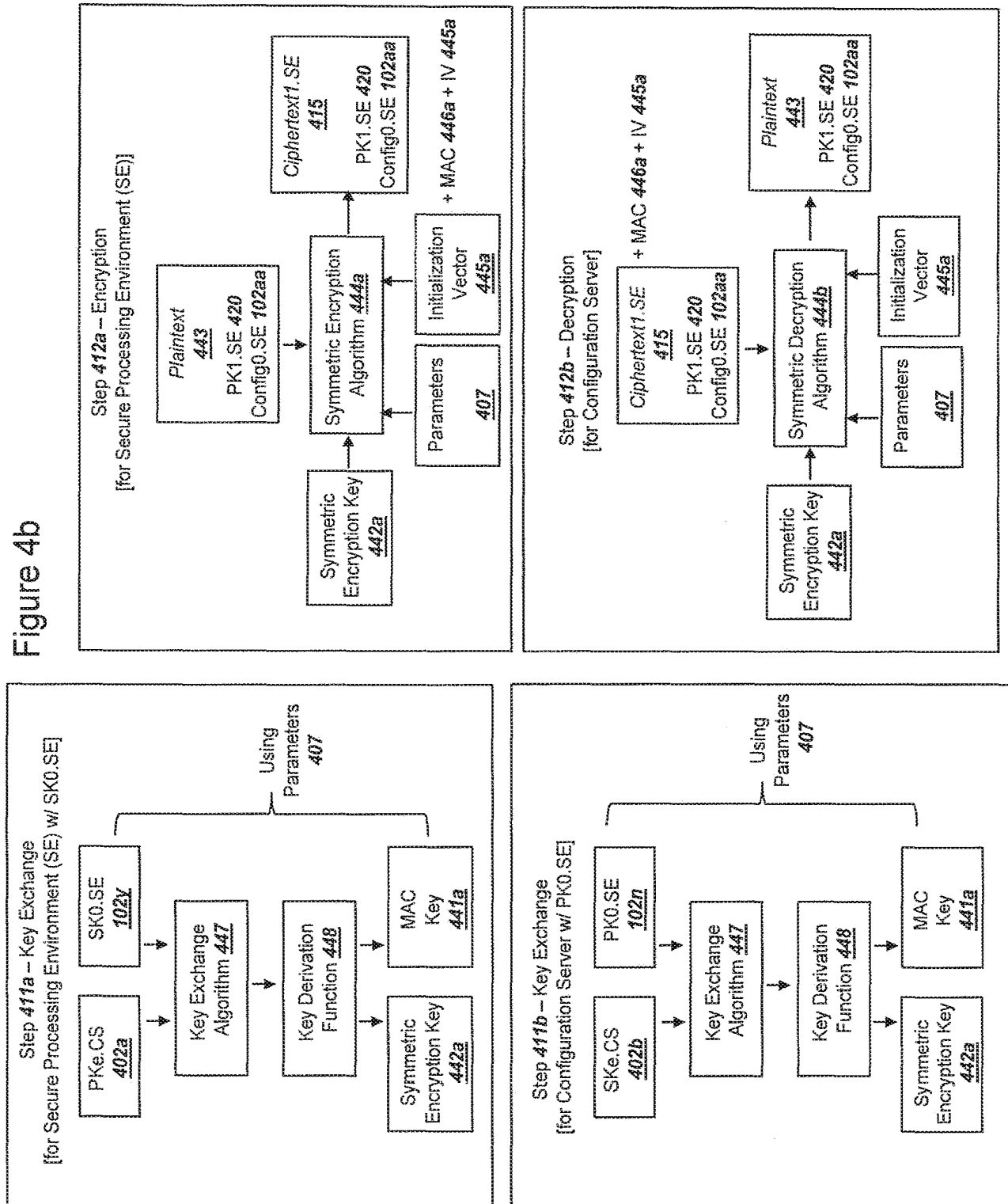


Figure 4c

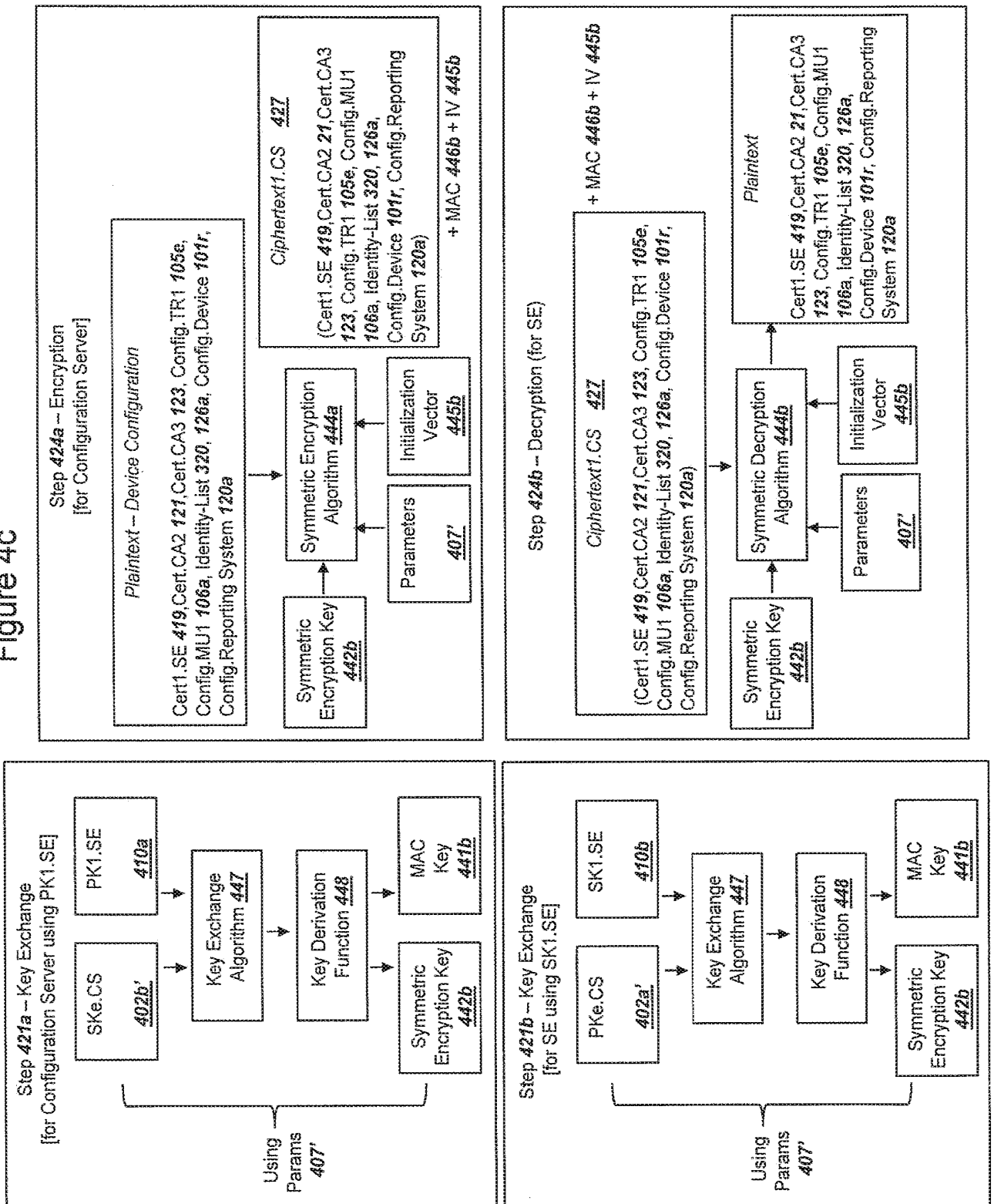


Figure 4d

SE Certificate cert1.SE **419** (with PK1.SE and Signed by CA1 115)

256 bit Elliptic Curve Key with SHA 256 Signature ("... " is long text removed)

```

Version: 3 (0x2)
Serial Number: 20486 (0x5006)
407 → Signature Algorithm: ecdsa-with-SHA256
Issuer: C=US, ST=Illinois, L=Chicago, O=Configuration System 114, ← 114
115 → OU=CA 115, CN=Certificate Authority 115 for Configuration System 114
/emailAddress=ca-115@configuration-system-114.org
Validity
Not Before: Aug 31 21:15:16 2017 GMT
407' → Not After: Aug 30 21:15:16 2022 GMT
Subject: C=US, ST=Illinois, L=Chicago, O=Device Manufacturer 101x,
O=JNIX, OU=Secure Processing Environment Type-Model,
102m → CN=SE ID 102m (Plus Optional Certificate Sequence Number)
Subject Public Key Info:
Public Key Algorithm: id-ecDH
Public Key: (256 bit) ← 407'
410a { 00:c6:f2:1f:b0:ec:78:ed:d5:5a:f9:c2:2b:03:54
...
39:05:82:41:6a:43:95:99:da:19:2e
407' → ASN1 OID: secp256r1
NIST CURVE: P-256
X509v3 extensions:
X509v3 Subject Key Identifier:
F9:03:6E:48:99:CE:04:AB:1C:6B:18:B1:CA:B2:3B:6A:19:4B:81:8D
X509v3 Authority Key Identifier:
keyid:46:AC:7F:83:C9:27: ... :A0:B8:E5:B3:1A:3A:48
DirName:/C=US/ST=Illinois/L=Chicago/O=Configuration System 114
/OU=CA 115/CN=Certificate Authority 115 for Configuration System 114
/emailAddress=ca-115@configuration-system-114.org
serial:02
X509v3 Key Usage:
Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment
X509v3 CRL Distribution Points:
URI:http://www.ca-115-for-configuration-system-114.org/ca/crl/ref_ca.crl
Signature Algorithm: ecdsa-with-SHA256
220 { 26:35:bb:aa:ff:29:7d:ee:3d:8d:e4:a3:b5:14:94:4d:13:ee:
...
c8:ba:f2:38:18:7d:81:0a:66:1e:8c:19:2a:cc:e5:ae:9c:32:
32:c3:5e:4d
-----BEGIN CERTIFICATE-----
MIIGMzCCBRugAwIBAgICUAYwDQYJKoZIhvcNEFBQAwwgaoxCzAJBgNVBAYTAIVT
...
5Ww3GBO1SxbgDihKnSGUAMXWY1laLXJSiKEGZSBmBBYlryOBh9gQpmHowZKszl
rpwyMsNeTQ==
-----END CERTIFICATE-----

```


Figure 6a

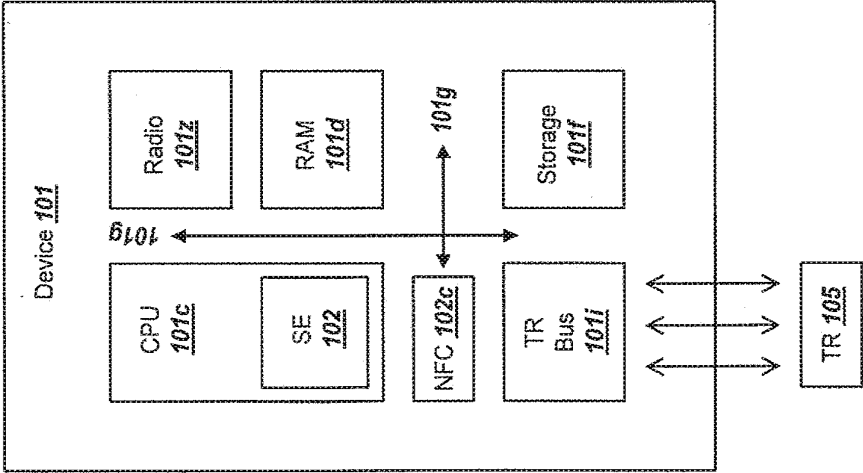
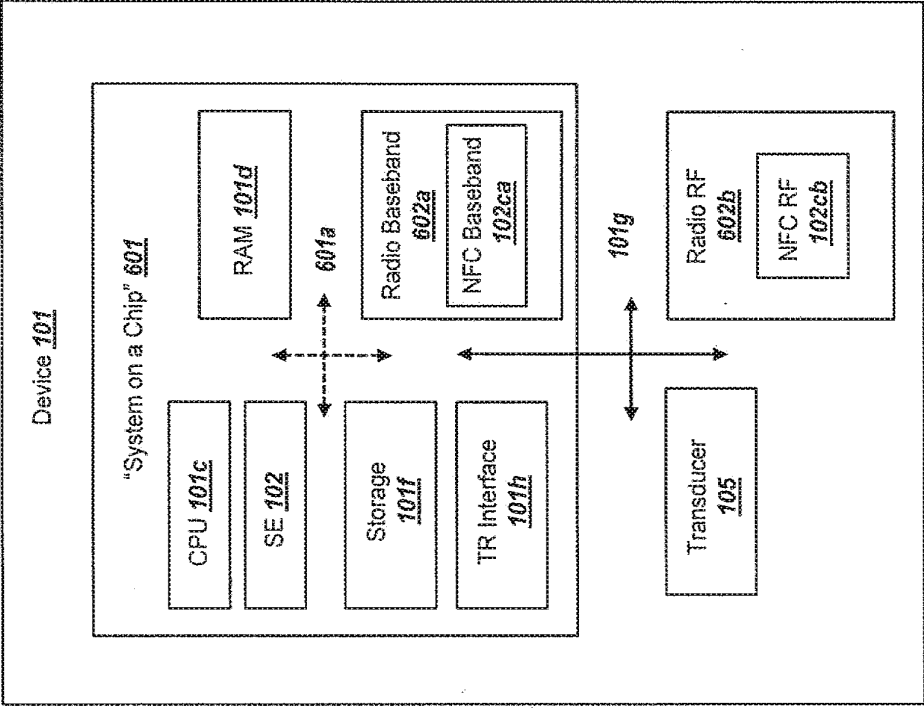


Figure 6b



INTERNATIONAL SEARCH REPORT

International application No

PCT/US2019/022184

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L29/06 H04L29/08 H04W12/04
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, COMPENDEX, INSPEC, IBM-TDB, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2017/339120 A1 (ZAKARIA OMAR [US] ET AL) 23 November 2017 (2017-11-23)	107-143
A	abstract	1-106
	paragraph [0035]	
	paragraph [0055] - paragraph [0056]	
	paragraph [0104] - paragraph [0114]	
	figures 1, 11, 12A, 12B	

A	US 2017/373845 A1 (NIX JOHN A [US]) 28 December 2017 (2017-12-28)	1-143
	the whole document	



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

16 May 2019

Date of mailing of the international search report

27/05/2019

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Horn, Marc-Philipp

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2019/022184

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2017339120 A1	23-11-2017	US 2017006003 A1	05-01-2017
		US 2017339120 A1	23-11-2017

US 2017373845 A1	28-12-2017	AU 2014342646 A1	19-05-2016
		CA 2965119 A1	07-05-2015
		GB 2534801 A	03-08-2016
		US 2015071139 A1	12-03-2015
		US 2015095648 A1	02-04-2015
		US 2015106616 A1	16-04-2015
		US 2015121066 A1	30-04-2015
		US 2015143125 A1	21-05-2015
		US 2015180653 A1	25-06-2015
		US 2015304113 A1	22-10-2015
		US 2016149709 A1	26-05-2016
		US 2016164678 A1	09-06-2016
		US 2016234020 A1	11-08-2016
		US 2016270000 A1	15-09-2016
		US 2017188231 A1	29-06-2017
		US 2017237561 A1	17-08-2017
		US 2017302447 A1	19-10-2017
		US 2017373845 A1	28-12-2017
		US 2018254897 A1	06-09-2018
		US 2018262329 A1	13-09-2018
		US 2018270059 A1	20-09-2018
		US 2018343117 A1	29-11-2018
		WO 2015065913 A1	07-05-2015



(51) **International Patent Classification:**
H04W 12/06 (2009.01) *H04W 12/04* (2009.01)

(21) **International Application Number:**
PCT/US2020/030424

(22) **International Filing Date:**
29 April 2020 (29.04.2020)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
62/841,435 01 May 2019 (01.05.2019) US

(72) **Inventor; and**

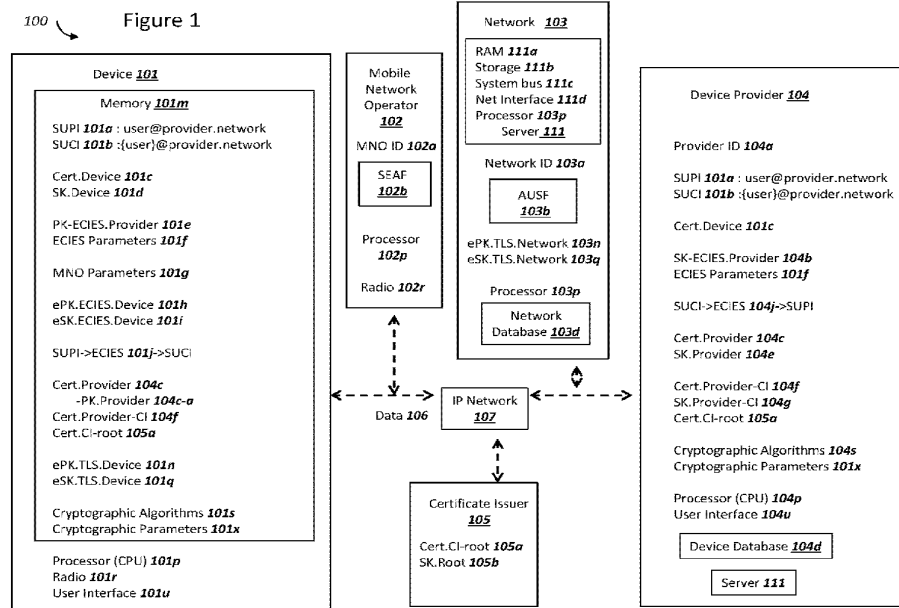
(71) **Applicant: NIX, John, A.** [US/US]; 807 Davis St., Unit 2207, Evanston, IL 60201 (US).

(74) **Agent: THYMIAN, Marcus, J.;** McDonnell Boehnen Hulbert & Berghoff LLP, 300 South Wacker Drive, Chicago, IL 60606 (US).

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ,

(54) **Title:** DISTRIBUTED EAP-TLS AUTHENTICATION FOR WIRELESS NETWORKS WITH CONCEALED SUBSCRIBER IDENTITIES



(57) **Abstract:** A device, mobile operator, network, and a device provider can exchange messages for EAP-TLS authentication. The network can include an authentication server function (AUSF). A device and device provider can record both a device certificate and a device provider certificate. The device provider can receive a subscriber concealed identity (SUCI) for the device from the AUSF. The device provider can decrypt the SUCI and send the network the certificates with a device SUPI. The network can send at least a TLS ephemeral public key to the device provider. The device provider can generate a digital signature for at least the ephemeral public key with a private key for the device provider certificate. The AUSF can complete an EAP-TLS authentication with the device using at least (i) the device certificate for the device, (ii) the device provider certificate for the server, (iii) the digital signature, and (iv) the ephemeral public key.

[Continued on next page]

UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

DISTRIBUTED EAP-TLS AUTHENTICATION FOR WIRELESS NETWORKS
WITH CONCEALED SUBSCRIBER IDENTITIES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Application No. 62/841,435, which is hereby incorporated herein by reference in its entirety.

BACKGROUND

Technical Field

The present systems and methods relate to conducting an extensible authentication protocol -- transport layer security (EAP-TLS) authentication with concealed subscriber identities for wireless networks, and more particularly to storing and using keys and certificates with a device, a network, and a device provider in order to improve efficiency, increase flexibility, and enhance security of the EAP-TLS authentication.

Description of Related Art

The use of EAP-TLS authentication has been widely adopted over the past decade within enterprise wireless networks using WiFi standards such as 802.11. Standards for EAP-TLS authentication include the Internet Engineering Task Force (IETF) Request for Comments (RFC) 5216 titled "The EAP-TLS Authentication Protocol", which is herein incorporated by reference in its entirety. Over the past two years, the European Technical Standards Institute (ETSI) has begun publishing standards for 5th generation (5G) wireless wide-area networks (WAN), such as described in TS 133 501 V15.1.0 titled "5G; Security architecture and procedures for 5G System", which is herein incorporated by reference in its entirety. Unlike prior 4th generation wireless WAN networks, 5G standard networks can include the support of authentication of devices using EAP-TLS, as described in Annex B of TS 133 501. EAP-TLS authentication enables the use of certificates for mutual authentication of a device and a wireless network. Previous wireless WAN standards for authentication such as with 4G LTE, 3G, etc. have depended on a pre-shared secret key recorded in a device and a secure server within the network to conduct an authentication and key agreement (AKA) protocol. The significant changes with 5G networks removes some previous limitations, but also creates additional needs in the art in order to efficiently support EAP-TLS authentication for thousands or more different types of devices connecting with hundreds or more different wireless networks globally.

The inclusion of support for EAP-TLS in 5G networks can ease the support for wireless device authentication and secure connectivity for many applications, where a pre-shared secret key such as a key K within a subscriber identity module (SIM) or embedded SIM may no longer be required. Most commercial arrangements for device owners or device users bind the use of the pre-shared secret key K within a device to a particular home wireless network that also records and derives keys from the same

pre-shared secret key. 5G networks also include the use of subscribed concealed identities as a partially encrypted form of a subscriber permanent identity. 5G networks as specified in TS 133 501 also include the use of a network access identifier (NAI) according to IETF RFC 7542 titled “The Network Access Identifier”, which is herein incorporated by reference in its entirety. In addition, in 2018 the IETF updated TLS version 1.2 to TLS version 1.3 in IETF RFC 8446 titled “The Transport Layer Security (TLS) Protocol Version 1.3”, which is herein incorporated by reference in its entirety. ETSI published standards as of May 2019 such as TS 133 501 for 5G networks do not address the use of TLS version 1.3 with EAP-TLS authentication. A need exists in the art for devices, wireless networks, networks, and device providers to support an EAP-TLS authentication using TLS version 1.3 with the encrypted or partially encrypted subscriber concealed identities as described in TS 133 501 Annex B, where the corresponding subscriber permanent identities can use the NAI format.

Although the use of PKI certificates and private keys for EAP-TLS authentication can avoid the logistical and provisioning complexities of using a pre-shared secret key for authentication (such as with 4G and 5G AKA authentication), the use of EAP-TLS authentication can create a new series of problems or needs in the art for the many different devices and wireless networks which use EAP-TLS authentication. These problems may be avoided or more easily addressed for existing implementations of EAP-TLS authentication with corporate or enterprise WiFi networks, for example, because the enterprise can control both the device and the wireless network. The enterprise can ensure (i) a root certificate or certificate issuer certificate stored in the wireless network can be used to authenticate the device and (ii) and a root certificate or certificate issuer certificate stored in the device can be used to authenticate the wireless network. Both (i) and (ii) in the previous sentence are often required for mutual authentication using EAP-TLS standards. In addition, TLS authentication and PKI certificates and secret keys can support many different sets of cryptographic algorithms and cryptographic parameters. An enterprise deploying EAP-TLS authentication currently can control the device certificates and the wireless network certificates before device deployment and installation to ensure the PKI certificates support compatible sets of cryptographic parameters and algorithms.

However, for widespread and even global deployments of 5G networks and devices, there can be hundreds or more different device manufacturers, device providers, and wireless networks, where each could use many different cryptographic algorithms, parameters, and/or certificate issuers. In addition, (A) the device manufacturers or device providers storing device default PKI certificates in devices using device default cryptographic algorithms and cryptographic parameters may not support (B) the network PKI certificates and network supported cryptographic algorithms and cryptographic parameters. For many applications, no pre-existing relationship may exist between a device manufacturer and a wireless network before device distribution to end users and the device's first connection to a 5G wireless network. A device, using EAP-TLS authentication and a subscriber concealed identity, may not record a certificate chain or root certificate that can be used to authenticate

a network certificate received from the network in a TLS handshake for EAP-TLS authentication. A need exists in the art for a wireless network to support EAP-TLS authentication for any one of hundreds or more of different device types using concealed identities from hundreds or more different device providers. A need exists in the art, where the device uses a subscriber concealed identity, to support verification of a certificate received from any one of hundreds or more different wireless networks, where no pre-existing relationship may exist between the device owner and the wireless network.

For example with the hundreds or more different 5G wireless networks globally and with a manufactured or installed set of certificate issuer certificates or root certificates, a device could receive a server certificate from an authentication server function (AUSF) that does not chain or may not be able to be verified through a root certificate or certificate issuer certificate recorded by the device. In addition, the server certificate may use cryptographic algorithms and parameters that are different than those used by a certificate issuer certificate stored within the device. The device may not have access to an IP network without first authenticating with a wireless network. This potentially creates significant problems, where the device does not have access to the IP network to receive (i) a supporting certificate for verifying the server certificate from (ii) a server other than through the wireless network where a connection is attempted. In other words, the supporting certificate may be required in order to verify the server certificate received from the wireless network, but the device may not be able to receive the supporting certificate from an IP network without first successfully completing an EAP-TLS authentication.

Similarly, (i) the wireless network may control or establish a first certificate chain for a server certificate used by an authentication server function (AUSF), (ii) a device provider may control or establish a second certificate chain within a device before distribution to an end user for verifying the server certificate, and (iii) the server certificate from a wireless network may not be verifiable by the second certificate chain. The problem of (iii) is compounded by the thousands or more potential combinations of different certificate issuer certificates and supported cryptographic parameters for required certificates to successfully complete an EAP-TLS authentication. A need exists in the art for ensuring a device can authenticate a server certificate from a wireless network using both (i) a certificate issuer certificate recorded in the device before distribution, and (ii) cryptographic algorithms and parameters recorded in the device before distribution that would support verification of the server certificate. A need exists in the art for a device to support EAP-TLS mutual authentication, without requiring a device user to load a server certificate or supporting certificate issuer certificate in order for the device to conduct mutual authentication with a wireless network. A need exists in the art for a device to record compatible cryptographic algorithms and cryptographic parameters for a server certificate without requiring a device user to load compatible cryptographic algorithms and cryptographic parameters that can be required in order for the device to use or verify the server certificate.

Without access to an IP network (such as after successfully completing EAP-TLS authentication with the wireless network), the device may not be able to request a server certificate that can be verifiable through an installed certificate chain in a device, and consequently mutual authentication using methods such as EAP-TLS may fail or require manual intervention or unnecessary interaction with a device user in order for the EAP-TLS authentication to succeed. Again, the above problems related to server certificate verification by a device and the use of different cryptographic algorithms and parameters for different server certificates received by a device is compounded by the fact that there could globally be hundreds of mobile network operators and thousands or more device providers.

A device provider may not know which of the hundreds of different mobile networks a particular device may try to connect with until the device is turned on by a user or first attempts to connect with a mobile network in a particular geographical region. Even if a particular geographical region is known, such as the United States, before a device attempts to connect to one of a handful of wireless networks, the preferred wireless network may not be known to a device provider before a device attempts to connect. A need exists in the art for a device to receive a server certificate in a TLS handshake from a wireless network supporting EAP-TLS, where verification of the server certificate does not require intervention from an end user, in order to allow a device to mutually authenticate with EAP-TLS using a subscriber concealed identity with a plurality of different mobile network operators.

In addition, a device provider may control or establish a certificate chain for verifying a device certificate stored within a device, and (ii) an authentication server function may record a set of certificate issuer certificates and/or root certificates with cryptographic algorithms and cryptographic parameters that may not support the verification of the device certificate. Devices from hundreds or more different device providers can also use different certificate chains for the device certificates and also different sets of cryptographic algorithms and cryptographic parameters. The request by the authentication server function for supporting certificate issuer certificates and/or cryptographic algorithms and cryptographic parameters for a received device certificate may take additional time, complexity, and potential security issues for securely receiving a supporting certificate issuer certificate and/or cryptographic algorithms and cryptographic parameters in order to properly verify a received device certificate. A need exists in the art for ensuring an authentication server function can securely and efficiently authenticate a device certificate received through a wireless network, where the device can utilize a subscriber concealed identity.

Many other examples exist as well for needs in the art for devices to securely and efficiently conduct EAP-TLS authentication with wireless networks, where the device uses a subscriber concealed identity, and the above are examples are just a few and intended to be illustrative instead of limiting.

SUMMARY

Methods and systems are provided for a device, a mobile network operator (MNO), a network, and a device provider to support EAP-TLS authentication, where the device can use a subscriber concealed identity. A system can include a plurality of devices, a mobile network operator, a network, and a device provider. The device can record a subscriber permanent identifier (SUPI), a device certificate, a device private key corresponding to the device public key in the certificate, an elliptic curve integrated encryption scheme (ECIES) function, a static provider public key for the ECIES function, MNO parameters, a device provider certificate, and cryptographic algorithms and cryptographic parameters used with both the device provider certificate and the device certificate. The subscriber permanent identifier can comprise a network access identifier (NAI) supporting IETF RFC 7542, where the NAI can include a user name and a realm. The realm can comprise a first portion with a provider identity and a second portion with a network identity.

The device can be a computing device for connecting with wireless networks such as a 5G network or a Wi-Fi network and the device can include a radio. The device can include a processor, memory, a user interface, and a bus connecting the components within an enclosure. The device could comprise a radio module within a larger device, such as a radio module within an automobile, industrial equipment, a mobile handset such as a smart phone, and other possibilities exist as well for physical embodiments of the device. The device can conduct an EAP-TLS authentication through the MNO with the network.

The mobile network operator can comprise a wireless network with geographically distributed radios and antennas in order to wirelessly connect with the plurality of devices. The MNO can include a MNO identity in order to uniquely identify the MNO, such as including a mobile country code and a mobile network code. The MNO can receive messages from the device through a wireless network and forward the messages to the network. The MNO can receive messages from the network and forward the messages to the device through the wireless network. The MNO can establish a secure session through an IP network with the network.

The network can include a plurality of servers operating in a coordinate manner to support communications with a device. The plurality of servers for the network can comprise a server system. The network can include an authentication server function (AUSF) and a network database. The network can include an identity comprising the network identity in the second portion of a realm for the device subscriber permanent identifier. The network can conduct an EAP-TLS authentication with the device through the MNO supporting 5G standard such as TS 133 501 V15.1.0. The network can establish a secure session with a device provider. The network can operate a network database for storing data associated with each of the plurality of different devices connecting with the network through the MNO.

The device provider can comprise a collection of servers for an entity that controls the device before the device connects with the MNO. The device provider can represent a device manufacturer, a device distributor, or a device owner for a plurality of different devices. The device provider can include an identity comprising the provider identity in the first portion of a realm for the device subscriber permanent identifier. The device provider can operate a device database storing data for the plurality of devices. The device provider can record the subscriber permanent identifier, a device certificate for the device, an elliptic curve integrated encryption scheme (ECIES) function for decryption of a subscriber concealed identity (SUCI), a static provider private key for the ECIES function, a device provider certificate, a device provider private key for the public key in the device provider certificate, and cryptographic algorithms and cryptographic parameters used with at least the device provider certificate.

The device can generate the subscriber concealed identity (SUCI) using at least the SUPI and the ECIES encryption function. The device can use the MNO parameters to establish a radio resource connection with the MNO. The device can send the SUCI to the MNO, and the MNO can use the plaintext second portion of the realm to select the network and send the SUCI to the network. The network can use the plaintext first portion of the realm to select the device provider and send the SUCI to the device provider. The device provider can use at least the ECIES function and the static provider private key for the ECIES function to decrypt the SUCI and read the SUPI. The device provider can use the SUPI to select a device provider certificate and a device certificate for the device from the device database. The device provider can send at least the device provider certificate and the device certificate to the network, along with the SUCI. The network can store the received device provider certificate and the device certificate along with the SUCI in the network database.

The network can send an EAP-TLS start command to the device through the MNO. The network can receive a "client hello" and the SUCI from the MNO, where the "client hello" can support TLS version 1.3 as specified in RFC 8446. The "client hello" can include an ephemeral elliptic curve cryptography (ECC) public key for the device, where the device stores the corresponding device ephemeral private key. The AUSF can generate a server ephemeral PKI key pair compatible with the received ECC public key from the device in the "client hello" message, including a server ephemeral public key. The network can store the PKI keys in the network database. The network can use the SUCI to select the device provider certificate received from the device provider in the paragraph above as a server certificate for conducting the TLS version 1.3 handshake. The network can use the SUCI received with the "client hello" to select the device provider certificate from the network database. The network can process a "server hello" using at least the selected device provider certificate. Data processed by the AUSF for a "server hello" can include standard data such as the server ephemeral public key, selected extensions, a certificate request, and the selected device provider certificate. The data processed in the previous sentence can be sent in a "message to sign" request to the device provider

from the network. The device provider can use the device provider secret key (which can correspond to the device provider public key in the device provider certificate) to digitally sign the message to sign using a digital signature algorithm and parameters specified in the device provider certificate. The resulting digital signature can comprise a "CertificateVerify" portion of a "server hello" message. The device provider can send the resulting digital signature to the network.

The network can generate a shared secret and a symmetric ciphering key using the device ephemeral public key received in the "client hello" and the derived server ephemeral private key with an elliptic curve Diffie-Hellman (ECDH) key exchange. The network can store the derived shared secret and the symmetric ciphering key with the SUCI for the device in the network database. The server can encrypt at least the selected extensions, a certificate request, and the selected device provider certificate and the "CertificateVerify" into a ciphertext using the derived symmetric ciphering key. The network can send the device a "server hello" message with the server ephemeral public key and the ciphertext through the MNO, where the message to the MNO can include the SUCI so that the MNO can route the message to the device.

The device can receive the "server hello" message and take steps to process the message. The device can generate the shared secret and the symmetric ciphering key using the server ephemeral public key received in the "server hello" and the derived device ephemeral private key using the ECDH key exchange. The device can decrypt the ciphertext using the symmetric ciphering key. The device can read the plaintext value for the device provider certificate. The device can verify that the received device provider certificate is the same as the device provider certificate recorded by the device before the device connected with the MNO. In this manner, the device provider certificate as used by the AUSF and MNO can be trusted without requiring the device to verify the received device provider certificate through a stored certificate issuer certificate. If the received device provider certificate is not the same as a recorded device provider certificate, then the device could verify the received device provider certificate through a certificate issuer chain stored within the device.

In other words, the device does not need Internet connectivity in order to verify the received device provider certificate as a server certificate in the "server hello" message because the device provider certificate can be verified with data previously stored inside the device before connecting with the MNO. The selection of the received device provider certificate above by the device provider (by sending the device provider certificate to the network) can ensure the device provider certificate can be verified by the device without the device needing connectivity to an IP network (in order to query for other intermediate verifying certificates), thereby solving some needs in the art identified above.

The device can also verify the "CertificateVerify" digital signature portion of the decrypted "server hello" ciphertext using the public key with the received device provider certificate. In this manner, the device can trust that the AUSF has access to the device provider secret key corresponding to the public key in the device provider certificate (although the device provider and not the AUSF may

store and operate with the device provider secret key). Verification of the "CertificateVerify" digital signature can comprise an authentication of the network and/or MNO with the device.

The device can use the device provider certificate and the user portion of the SUPI to select a device certificate stored in the device. For example, the device could record multiple device certificates and select a device certificate compatible with the cryptographic parameters and algorithms within the device provider certificate. Or the device could select the device certificate based on the extensions within the decrypted ciphertext within the "server hello" message. The device can generate a response to the "server hello", where the response includes the selected device certificate and a digital signature or "CertificateVerify" from the device. The device could generate the "CertificateVerify" using the device private key corresponding to the device public key in the device certificate selected by the device. The device can encrypt the response using at least the mutually derived shared secret and data from the "client hello" and "server hello" messages. The device can send the encrypted data in the response to the MNO. The MNO can forward the response with the SUCI to the network.

The network/AUSF can use the SUCI received with the encrypted response to select the mutually derived shared secret and/or the symmetric ciphering key from the network database. The network/AUSF can decrypt the encrypted response from the device using at least the mutually derived shared secret to read the plaintext values of the device certificate and the digital signature for the device "CertificateVerify". The AUSF can compare (i) the received device certificate from the device with (ii) the device certificate stored in the network database, where the stored device certificate in the network database could be received previously as described above from the device provider. If (i) the received device certificate from the device and (ii) the received device certificate from the device provider are the same, then the device certificate can be trusted or considered verified. Or, if the device certificate received from the device is not the same as the device certificate received from the device provider, then the AUSF can conduct a certificate verification for the device certificate received from the device through a chain of certificate issuer certificates. The AUSF can verify the device "CertificateVerify" in the response by generating the equivalent of a digital signature or a hash value using the device public key from the device certificate, and if the received digital signature is equal to or the same as the generated digital signature, then the device can be considered authenticated with the network.

The AUSF can use (i) the shared secret key from the TLS ECDH key exchange stored in the network database for the device and (ii) the plaintext data from the "client hello" and the "server hello" in order to generate a TLS master secret according to the steps in IETF RFC 8446, such as section 7.1 for the key schedule. The AUSF can use the generated or derived TLS master secret to generate at least a value for an extended master secret key (EMSK) as specified in EAP-TLS standards, such as inputting the TLS master secret into a key derivation function that could include a secure hash algorithm. The AUSF can use a portion of the EMSK, such as the leading 256 bits from an EMSK as an anchor key or Kausf. The AUSF can send the anchor key along with the SUPI for the device to the MNO. The MNO

can use the anchor key to derive subsequent keys required for secure wireless communications with the device.

The device can conduct the equivalent steps of using the shared secret key and the plaintext data from the “client hello” and the “server hello” in order to generate the TLS master secret according to the steps in IETF RFC 8446. The device can use the generated or derived TLS master secret to generate at least a value for EMSK as specified in EAP-TLS standards, such as inputting the TLS master secret into a key derivation function that could include a secure hash algorithm. The device can use a portion of the EMSK, such as the leading 256 bits from an EMSK as an anchor key or Kausf. The device can use the anchor key to derive subsequent keys required for secure wireless communications with the mobile network operator.

These as well as other aspects and advantages will become apparent to those of ordinary skill in the art by reading the following detailed description, with reference where appropriate to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Various exemplary embodiments are described herein with reference to the following drawings, wherein like numerals denote like entities.

Figure 1 is a graphical illustration of an exemplary system, where a device communicates data with a mobile network operator and a network in order to mutually authenticate with the network, in accordance with exemplary embodiments;

Figure 2 is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a mobile network operator, a network, and a device provider, in accordance with exemplary embodiments; and

Figure 3 is an illustration of a device database and a network database with exemplary data stored, in accordance with exemplary embodiments.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS**Figure 1**

Figure 1 is a graphical illustration of an exemplary system, where a device communicates data with a mobile network operator and a network in order to mutually authenticate with the network, in accordance with exemplary embodiments. The system 100 can include a device 101, a mobile network operator 102, a network 103, a device provider 104, and a certificate issuer 105. The depicted nodes or entities can communicate data 106 over an Internet Protocol (IP) network 107. Although a single device 101, a single mobile network operator 102, a single network 103, a single device provider 104, and a single certificate issuer 105 are depicted in Figure 1, a system 100 can comprise a plurality of each of the depicted nodes or entities. A system 100 as depicted in Figure 1 can support EAP-TLS authentication for 5G wireless WAN networks, such as supporting EAP-TLS authentication as described in Annex B of TS 133 501 V15.1.0, although other wireless networking technology for device 101 and MNO 102 could be supported as well.

Device 101 and mobile network operator 102 can utilize a variety of wireless wide area network (WAN) and wireless local area network (LAN) wireless and technologies to communicate data 106 between the nodes, including Low Power Wide Area (LPWA) technology, 3rd Generation Partnership Project (3GPP) technology such as, but not limited to, 3G, 4G Long-Term Evolution (LTE), or 4G LTE Advanced, NarrowBand – Internet of Things (NB-IoT), LTE Cat M, and 5G or subsequent wireless technologies. In addition, the wireless technology used by device 101 and mobile network operator 102 could support or implement wireless LAN technologies such as WiFi and the related series of standards from IEEE 802.11 standards, such as 802.11ac, 802.11ax, etc. Other examples exist as well for wireless WAN technology and/or wireless LAN technology used by device 101 and mobile network operator 102 without departing from the scope of the present disclosure.

Mobile network operator 102, network 103, and device provider 104 can connect to the IP network 107 and communicate with each other via a wired connection such as, but not limited to, an Ethernet connection, or a fiber optic connection. In other words, mobile network operator 102 can connect to (i) device 101 through wireless technology and (ii) a network 103, a device provider 104, and a certificate issuer 105 using wired technology. IP network 107 could also be a public or private network supporting Internet Engineering Task Force (IETF) standards such as, but not limited to, such as, RFC 786 (User Datagram Protocol), RFC 793 (Transmission Control Protocol), and related protocols including IPv6 or IPv4. A public IP network 107 could utilize globally routable IP addresses. A private IP network overlayed on IP network 107 could utilize private IP addresses which could also be referred to as an Intranet. Other possibilities for IP Network 107 and a private network between mobile network operator 102, network 103, and device provider 104 exist as well without departing from the scope of the disclosure.

Mobile network operator (MNO) 102 can include a plurality of radio-frequency (RF) access technologies (RAT) systems for supporting wireless communications with a plurality of devices 101 in a networked manner, including through the use of geographically dispersed antennas, radio nodes, or towers. The RAT systems for MNO 102 can include a radio 102r for communicating with device 101, where radio 102r includes an antenna system and can operate through licensed radio spectrum. In exemplary embodiments, MNO 102 can operate a plurality of radios 102r which are connected to an IP network 107 in a secure manner, including connecting via a private IP network.

In exemplary embodiments, mobile network operator 102 can comprise traditional public land mobile networks providing data and voice services such as AT&T, Verizon, T-Mobile, Sprint, etc, and provide data communications services through a variety of radio access technologies. Further, although Figure 1 depicts an MNO 102 as communicating with device 101 with a radio 102r, another entity besides a mobile network operator could perform the function of MNO 102. For embodiments where radio 101r for device 101 uses WiFi technology, then MNO 102 could comprise an entity other than a PLMN, such as a service or network that operates a plurality of WiFi access points. The function of MNO 102 could also be conducted by a large enterprise with a collection of geographically distributed WiFi access points.

Each of mobile network operator (MNO) 102, network 103, and device provider 104 could operate a plurality of servers in order to support the communications and connectivity depicted in Figure 1 and also Figure 2 below. The exemplary data structures, values or numbers, and steps for a MNO 102, network 103, and device provider 104 depicted in Figure 1 and also Figure 2 below could be recorded and/or conducted by a collection of servers for each entity. Exemplary servers for a mobile network operator 102, network 103, or device provider 104 in system 100 can be either different physical computers such as rack-mounted servers, or different logical or virtual servers or instances operating in a “cloud” configuration.

An exemplary server or collection of servers for MNO 102 can comprise a Security Anchor Function (SEAF) 102b, where a MNO 102 can include at least one SEAF 102b for establishing secure and authenticated communications with device 101 through at least one radio 102r. The SEAF 102b could comprise and operate according to the European Technical Standards Institute (ETSI) standard TS 133 501 V15.1.0 from August of 2018 and titled “5G; Security architecture and procedures for 5G System”, which is hereby incorporated by reference in its entirety. In exemplary embodiments, the SEAF 102b for MNO 102 can receive an anchor key from network 103 in order to establish secure and authenticated communication with device 101, such as depicted and described for a message 237 below in Figure 2. In exemplary embodiments where Extensible Authentication Protocol – Transport Layer Security (EAP-TLS) in a system 100, then SEAF 102b could operate as specified in section 6.1.1.2 of TS 133 501 V15.1.0, where “the SEAF takes the role of pass-through authenticator.”

Note that MNO 102 could use other wireless networking technologies besides 5G. Thus, although an SEAF 102b is depicted in Figure 1, a different collection of servers or an entity besides a SEAF 102b could perform the steps for a MNO 102 depicted in Figure 2 below, without departing from the scope of the present invention. In general, the function of an SEAF 102b as depicted and described in connection with Figure 2 below could be conducted by at least one server or process operating within a cloud computing environment, where the server or process may not be an SEAF 102b. For example, the server or process operating with equivalent functionality as an SEAF 102b in Figure 2 below could (i) receive a partially encrypted Network Access Identifier (NAI) such as a SUCI 101b from device 101 with an encrypted user name and a plaintext realm for the NAI, and (ii) use the plaintext realm to forward the SUCI 101b to network 103, where network 103 uses or is associated with at least a portion of the plaintext realm.

MNO 102 can include a MNO identity 102a, where MNO identity 102a can comprise the combination of a mobile country code and a mobile network code, according to 3GPP standards for 4G networks, 5G networks, etc. As one example, MNO identity 102a for an example network of AT&T could comprise an MCC number of 310 and a MNC number of 410, such that the MNO identity 102a can be 310410. Note that a MNO 102 can operate with multiple MNO identities 102a, and a MNO identity 102a could also comprise a Domain Name Service (DNS) name such as an exemplary value of “att.net”, and other possibilities exist for a MNO identity 102a without departing from the scope of the present invention. In addition, a MNO 102 could operate with a plurality of MNO identities 102a.

MNO 102 can include a plurality of processors 102p in order to store and record data as well as communicate within a system 100 over an IP network 107 and a radio 102r. Processor 102p can comprise a general purpose processor appropriate for the computational requirements for a MNO 102, and may operate with multiple different processor cores, including field programmable gate arrays (FPGA). A processor 102p can comprise exemplary ARM® based processors or an Intel® based processor such as belonging to the XEON® family of processors, and other possibilities exist as well. Processor 102p can utilize a data bus to fetch instructions from memory or storage within a server and operate on the instruction. A processor 102p can include components such as registers, accumulators, and logic elements to add, subtract, multiply, and divide numerical values and record the results in memory. In exemplary embodiments, at least one processor 102p within MNO 102 can be used with an SEAF 102b to conduct the steps for MNO 102 and send/receive messages for MNO 102 as depicted and described in connection with Figure 2 below. Although not depicted in Figure 1, MNO 102 can include memory and at least one database in order to establish communications with a plurality of devices 101 through a plurality of radios 102r.

Device 101 can be a computing device for sending and receiving data using a radio 101r. Device 101 can take several different embodiments, such as a general purpose personal computer, a mobile phone or mobile handset based on the Android® or Fuchsia from Google® or the IOS operating system

from Apple ®, a tablet, a device with a sensor or actuator for the “Internet of Things”, a module for “machine to machine” communications, a device that connects to a wireless Wide Area Network (WAN) operated by MNO 102, a router, and/or a server, and other possibilities exist as well for the embodiments of a device 101 without departing from the scope of the present invention.

The electrical components within device 101 can include a memory 101m, a processor 101p, a radio 101r, and a user interface 101u. Although not depicted in Figure 1, a data bus or a system bus could internally electrically connect the depicted components within a device 101. Additional components to support the operation of device 101 can include a battery to store electrical power, an antenna to transmit and receive RF signals, a sensor to collect data external or internal to the device 101, such as temperature, motion, position, pressure, etc. A device 101 could also include an actuator to convert electrical signals into physical actions, such as a motor for moving components, a relay for opening or closing a circuit, etc.

Memory 101m can comprise combinations of (i) volatile random access memory and nonvolatile memory. The volatile memory can include random access memory (RAM) for relatively fast read and write operations compared to nonvolatile memory, such as SRAM or DRAM. RAM for memory 101m could also include persistent RAM or nonvolatile RAM (NVRAM), such that data in a persistent RAM memory or nonvolatile RAM is stored when power is removed. Nonvolatile memory can include storage memory such as a flash memory and record or store data when power is removed from device 101. In general, different forms and electrical components for memory 101m can be used without departing from the scope of the present disclosure. Processor 101p can comprise a central processing unit (CPU) or a “system on a chip” and be similar to a processor 102p for a MNO 102 described above, but with reduced capabilities for a device 101 compared to a processor 102p for MNO.

Device 101 may include radio 101r support radio-frequency (RF) communications with networks including a MNO 102 via standards such as GSM, UMTS, mobile WiMax, CDMA, LTE, LTE Advanced, 5G, and/or other mobile-network technologies. In a wireless configuration, the radio 101r may also provide connectivity to local networks such as 802.11 WLAN, Bluetooth, Zigbee, or an IEEE 802.15.4 network, among other possibilities. In exemplary embodiments, a radio 101r is connected to an antenna, which could be either internal to device 101 or external to device 101.

Note that device 101 may also optionally include user interface 101u which may include one or more devices for receiving inputs and/or one or more devices for conveying outputs. User interfaces are known in the art and thus user interfaces are not described in detail here. User interface 101u could comprise a touch screen if device 101 operates as a smartphone or mobile phone. Device 101 can optionally omit a user interface 101u, since no user input may be required for many M2M applications such as networked sensors, although a user interface 101u could be included with device 101. LED lights or a display of LEDs could also comprise a user interface 101u.

Device 101 can include a device identity comprising a subscriber permanent identity (SUPI) 101a. The device identity can comprise a string or number to uniquely identify device 101 with MNO 102, network 103, and device provider 104, and other nodes connected to the IP network 107. A device identity of a SUPI 101a can include a network access identifier (NAI) according to IETF RFC 7542, titled “The Network Access Identifier”, which is herein incorporated by reference in its entirety. A NAI can consist of different portions, such as a user identity, and a realm. The user identity can comprise a number or a string to uniquely identify device 101 within a realm. A NAI for a SUPI 101a can have many forms, and with several examples for a NAI are included in RFC 7542. Other possibilities exist as well for a NAI as a SUPI 101a for an identity of device 101 without departing from the scope of the present invention. The use of a SUPI 101a in an NAI format as a device identity for device 101 is not required for some exemplary embodiments, and for some embodiments the SUPI 101a could be a single string with an embedded user identity and realm but without the “@” character standard for a NAI.

As depicted in Figure 1, in exemplary embodiments, a SUPI 101a can have a user identity and a realm, where the realm can have a prefix and a suffix. The prefix and the suffix can have multiple portions or more than one portion. In general, (i) a realm suffix can include a base string of a domain name associated with a network 103 (depicted as “network” for SUPI 101a in Figure 1), and (ii) a realm prefix can include a second string associated with a device provider 104 (depicted as “provider” for SUPI 101a). In exemplary embodiments, the realm prefix can include at least a portion of a provider identity 104a and the realm suffix can include at least a portion of a network identity 103a. Note that a realm suffix could also include data or a name for MNO 102, which could include a MNO ID 102a. Other embodiments of a SUPI 101a for a device 101 than that depicted in Figure 1 could include exemplary values of “user@network.provider” (e.g. a prefix first portion of the realm includes a name for the network 103 and a suffix second portion of the realm includes a name for the device provider 104). In other exemplary embodiments, the realm of a SUPI 101a can comprise solely a name or identity for device provider ID 104a, and for these embodiments, MNO 102 and network 103 could have agreements and technical arrangements to route data 106 with the realm of a provider ID 104a to device provider 104.

In preferred embodiments, a device 101 with a user identity of “XXXXXX” could have be associated with a device provider with the name “Company”, where the company has a commercial agreement and technical agreement with a cloud service such as Amazon Web Services to operate the authentication server function (AUSF) 103b, such that devices 101 for the company (e.g. a device provider) could roam and/or be authenticated with a plurality of mobile network operators. The cloud service could have commercial and technical agreements with a plurality of mobile network operators, including MNO 102 depicted in Figure 1. The cloud service could have an exemplary network ID 103a of aws.com. For this exemplary embodiment described in this paragraph, the SUPI 101a for device 101 could comprise an example string of “XXXXXX@company.aws.com”. Note that an identity of MNO

102 such as MNO ID 102a may not need to be included in SUPI 101a, and MNO 102 can select network 103 based on the suffix of the realm (e.g. aws.com). In another embodiment, the MNO 102 identity could be included in the realm, where the MNO ID 102a could be “att.com”, and the SUPI 101b for device 101 could be “XXXXXX@company.aws.att.com”. Other possibilities exist as well without departing from the scope of the present invention.

Although not depicted in Figure 1, a NAI as a SUPI 101a could comprise a string where realm has an identity of the network 102a in the prefix and an identity of the device provider 103a in the suffix. Or, the realm could comprise a single base string that is unique for a combination of an identity of device provider 104a and network 102a (such as a realm of “company-aws.com”). Further, in some exemplary embodiments, the realm of a NAI for SUPI 101a could omit an identity device provider 104a and consist only of an identity for network 103 of network ID 103a (where the realm could be uniquely associated with a device provider 104). Or, the realm of a NAI for SUPI 101a could omit an identity network 102a and consist only of an identity for device provider 103 of provider ID 103a. In general, the realm of a SUPI 101a can include sufficient identifying information (such as at least a portion of network ID 103a and/or provider ID 104a) in order for data within a message from device 101 to be (i) received by MNO 102 and forwarded to network 103, and (ii) received by network 103 and forwarded to device provider 104.

A device 101 can also include a subscriber concealed identity (SUCI) 101b. SUCI 101b can include an encrypted identity of a user name or user identity, depicted as “{user}” in Figure 1 and the plaintext realm, where exemplary formats or contents of a realm were depicted and described in the two paragraphs above. A SUCI 101b can be derived from a SUPI 101a using an elliptic curve integrated encryption scheme (ECIES) 101j, which is described in addition detail below. In exemplary embodiments, device 101 communicates with a MNO 102 using a SUCI 101b as the identity for device 101 instead of the SUPI 101a, in order for the user identity of device 101 to be encrypted.

Although “user” is the term depicted in Figure 1 and also for a NAI in RFC 7542 and related 3GPP specifications, the “user” portion of a NAI can comprise a string of digits or numbers or characters to uniquely identify device 101 for a device provider 104 and the “user” portion of a NAI within a SUPI 101a or SUCI 101b does not need to be associated with a user such as a person. In other words, a “user” within an identifier such as a NAI or a SUPI 101a can be an identity for a device. The plaintext for the realm in a SUCI 101b can be required in order for MNO 102 and/or network 103 to route data for a SUCI 101b forward (e.g. MNO 102 -> network 103 and network 103-> device provider 104). MNO 102 could use a first plaintext prefix with network ID 103a in the realm to select a network 102. In other words, in exemplary embodiments, the realm portion of a NAI for a SUCI 101b can remain plaintext, although the “user” portion of a NAI for a SUCI 101b can be encrypted.

The “user” portion of a NAI for a SUPI 101a could comprise a medium access control (MAC) address for a physical interface such as Ethernet or WiFi, an international mobile subscriber identity

(IMSI) or international mobile equipment identity (IMEI) with 2G/3G/4G networks, and other possibilities exist as well for the user portion of a NAI as a SUPI 101a without departing from the scope of the present invention. In exemplary embodiments, SUPI 101a can be written to hardware in device 101 and operate as a globally unique, long-term identity for device 101. Note that a device 101 could use several different SUPI 101a, such as a first SUPI 101a with a first network 103 and a second SUPI 101a with a second network 103, where the realm portion can be different for the first and second SUPI 101a and the different realm portions can include different network IDs 103a.

A certificate for device 101 can comprise cert.device 101c and can include the device identity comprising the SUPI 101a. For some embodiments, the device identity within a device certificate cert.device 101c can be a number, string, or value that is different than the SUPI 101a. As one example, a SUPI 101a could be for connecting with a wireless network 102, while a device identity in a cert.device 101c can be for communicating with a device provider 104. A certificate 101c for device 101 can include a device public key corresponding to the device private key SK.Device 101d, a set of cryptographic parameters for the device public key, the identity of a certificate authority, and a digital signature for the certificate authority. A certificate 101c for device 101 and other nodes in a system 100 can comprise a certificate supporting X.509 v3 formats. The cryptographic parameters can specify parameters for both the device public key and also the digital signature from a certificate authority. Cryptographic parameters in a certificate can specify algorithms such as RSA or elliptic curve cryptography (ECC) values, ECC curve names, key lengths in bits, signature algorithms, validity dates, encoding rules, etc.

The signature of the certificate authority for cert.device 101c could be included in device certificate 101c. A node of system 100 in Figure 1, such as the authentication server function (AUSF) 103b can use the device certificate cert.device 101c in order to authenticate device 101 with MNO 102. Additional details regarding the use a certificate for device 101 of cert.device 101c will be described below in Figure 2. Note that other nodes in a system 100 can record and operate with a certificate similar to cert.device 101c, except with identities for the nodes, the public keys for the nodes, and certificate authority digital signatures for the public keys. A device 101 could record an use multiple different device certificates 101c (possibly with different device public keys corresponding to different device private keys), where the device 101 selects the different device certificates 101c based on the MNO 102 and/or the network 103 the device 101 communicates with.

A device 101 can record and use a device private key SK.device 101d (e.g. "secret key"). The private key can correspond to the public key in the device certificate cert.device 101c. In exemplary embodiments, device 101 internally generates a random number for the device private key SK.device 101d and using cryptographic algorithms 101s and generates the device public key in a certificate 101c. The private key SK.device 101d and the corresponding public key can be for the set of cryptographic parameters within the certificate 101c, such as the values for an elliptic curve named curve. The

generation of the public key for device certificate cert.device 101c could be (i) during device 101 manufacturing or distribution and (ii) before device 101 sends a SUCI 101b to a MNO 102.

In some exemplary embodiments before device 101 establishes a radio connection with MNO 102 (such as during a device configuration step), device provider 104 receives the public key for SK.device 101d and signs the public key with a device provider secret key such as SK.Provider 104e in order to generate the device certificate cert.device 101c. Or, a device certificate cert.device 101c could be signed by a different certificate issuer than a certificate issuer operated by device provider 104. Note that cryptographic algorithms 101s and PKI keys for device 101 can support post-quantum cryptography (PQC) and support code-based, lattice-based, and related cryptographic algorithms. Example possible algorithms for parameters in device certificate cert.device 101c include those described in "Post-Quantum Cryptography Standardization" from Wikipedia dated March 12, 2019, which is herein incorporated by reference.

Device 101 can store or record a public key for the device provider 104 of PK-ECIES.Provider 101e in order to conduct ECIES 101j encryption of the user name for a SUCI 101b. The public key can be associated with a set of cryptographic parameters ECIES Parameters 101f for using the public key (such as an ECC curve name), and the cryptographic parameters could be used with ECIES 101j function. In other words, ECIES Parameters 101f can specify a named curve associated with PK-ECIES.Provider 101e. The device provider 104 can record a private key corresponding to PK-ECIES.Provider 101e, where the private key can comprise SK-ECIES.Provider 104b as depicted in Figure 1. The value of PK-ECIES.Provider 101e could be written to memory 101m during manufacturing or distribution or configuration of device 101. Note that PK-ECIES.Provider 101e can comprise multiple different points on different ECC curves, such that different values of PK-ECIES.Provider 101e can be selected depending on the parameters ECIES Parameters 101f used by ECIES 101j. In other words, PK-ECIES.Provider 101e can include a first public key for name curve P-256, a second public key for named curve 25519, where the selection of the first public key or second public key could be based on ECIES Parameters 101f.

MNO parameters 101g can include a list of numbers or strings for values such as (i) allowed frequencies or frequency bands to scan, (ii) preferred access lists for roaming onto other wireless networks, (iii) criteria for a device 101 to select MNO 102, including in idle mode, (iv) support for emergency services, (v) supported languages or character encoding, (vi) codes to search for in beacons broadcast by a wireless network 102, (vii) parameters for a radio 101r to use when connecting to a wireless network 102, (viii) names or addresses for a server associated with a MNO 102 in order for a device 101 to conduct radio resource connect procedures, etc. A device 101 can record multiple sets of MNO parameters 101g for a plurality of different MNO 102, and a device 101 could select a MNO 102 from a plurality of available MNO 102 for radio 101r based on data within MNO parameters 101g.

Device 101 can derive an ephemeral public key ePK.ECIES.Device 101h and an associated secret key or private key eSK.ECIES.Device101i, using a random number, a key pair generation function, and the set of ECIES Parameters 101f. Device 101 can utilize a random number generator similar such as hardware random number generator in order to derive the private key eSK.ECIES.Device101i and input the random number into the key pair generation function. A key pair generation function could be included in application software running on device 101, and exemplary software for key pair generation algorithm can comprise OpenSSL, crypto++, or Mozilla libraries. The resulting ephemeral public key ePK.ECIES.Device 101h can be sent to MNO 102 in a message 205 below in Figure 2. The resulting secret key or private key eSK.ECIES.Device101i can be input into the encryption function ECIES 101j in order to convert a SUPI 101a into a SUCI 101b, where the user portion of an NAI for a SUPI 101a can be encrypted in a SUCI 101b. In some embodiments, the realm portion of the NAI in a SUCI 101b can remain plaintext, such that wireless network 102 can read the plaintext realm for the SUCI 101b in order to select and forward the SUCI 101b to a network 103 and/or device provider 104.

Device 101 can include an encryption function ECIES 101j in order to convert a SUPI 101a into a SUCI 101b. The encryption function ECIES 101j can comprise the identity encryption scheme as described in Annex C: "Protection schemes for concealing the subscription permanent identifier" in ETSI standard TS 33.501 version 15.1.0. In summary, device 101 can conduct an elliptic curve Diffie-Hellman key exchange using private key eSK.ECIES.Device101i and public key PK-ECIES.Provider 101e and parameters ECIES Parameters 101f in order to generate an encryption key and a message authentication code (MAC) key. The user portion of SUPI 101a, or the user name, can be input into the encryption function ECIES 101j along with the derived encryption key and MAC key in order to generate SUCI 101b which contains the user name as a ciphertext (depicted as "{user}" in Figure 1 and Figure 2 below). A MAC code generated by ECIES 101j using the MAC key can be used to verify message integrity for the SUCI 101b by device provider 104. The realm portion of SUPI 101a can remain as plaintext for SUCI 101b after encryption function ECIES 101j.

Device 101 can record or store a certificate for device provider 104 comprising Cert.Provider 104c, where the certificate for device provider 104 can be similar to the certificate for device 101 described above. An identity for device provider 104 recorded with Cert.Provider 104c can comprise provider ID 104a. Cert.Provider 104c can also record a public key for device provider 104 comprising PK.Provider 104c-a. Cryptographic parameters can be included in the certificate for device provider 104, such as an ECC named curve for PK.Provider 104c-a, a time for validity of the certificate, and other information normally recorded in an X.509 v3 certificate. Note that in exemplary embodiments, the certificate Cert.Provider 104c can be associated with and generated by device provider 104 and not MNO 102 and network 103. As depicted and described below in connection with Figure 2, exemplary embodiments contemplate that (i) device 101 can authenticate with network 103 and (ii) MNO 102 can

obtain an anchor key for device 101 without device 101 receiving or storing a certificate for MNO 102 and network 103 different than the stored or recorded device provider certificate Cert.Provider 104c .

In this manner, the use of a device 101 in a system 100 can be significantly more flexible than conventional technology, since a device provider 104 can control the device 101 before connecting to MNO 102 or selecting MNO 102 and device 101 does not need to receive a separate server certificate that chains to (or can be verified through) a previously recorded root certificate Cert.CI-root 105a. In other words, without the technology described herein, an MNO 102 and network 103 operating an AUSF 103b may not readily guarantee (or without user interaction for device 101) that device 101 can verify a server certificate received from AUSF 103b that can be verified with a recorded root certificate Cert.CI-root 105a.

The certificate Cert.Provider 104c can be signed by a certificate authority such as a device provider certificate issuer, and device 101 can record a chain of certificate authority certificates up to a root certificate Cert.CI-root 105a for a certificate issuer 105. For example, as depicted in Figure 1, device 101 Cert.Provider 104c could be signed by a certificate issuer for device provider 104 with a certificate comprising Cert.Provider-CI 104f. The public key for the device provider certificate issuer in Cert.Provider-CI 104f could be signed by the certificate issuer 105. The certificates for device provider 104 including Cert.Provider 104c and the supporting chain of certificates including Cert.Provider-CI 104f and root certificate Cert.CI-root 105a could be written by device provider 104 during distribution or configuration of device 101 or before device 101 is received by an end user of device 101. The certificates for device provider 104 including Cert.Provider 104c and the supporting chain of certificates including Cert.Provider-CI 104f and root certificate Cert.CI-root 105a could be stored by device 101 before device 101 sends data 106 to a MNO 102 in order to authenticate the device 101 with MNO 102 .

For embodiments where the certificate Cert.Provider 104c is written to device 101 or stored by device 101 before a device 101 connects with a MNO 102, then a device 101 could also (i) receive Cert.Provider 104c in a message from network 103 and (ii) check that the received Cert.Provider 104c matches the stored Cert.Provider 104c, and in this manner Cert.Provider 104c could be authenticated without requiring device 101 to conduct a full set of certificate signature verification steps through parent certificates of Cert.Provider 104c up to root certificate Cert.CI-root 105a.

Note that without recording Cert.Provider 104c in memory before connecting to a MNO 102, then device 101 could receive a server/network certificate from AUSF 103b that does not chain to (or be verified through) root certificate Cert.CI-root 105a, which can cause significant problems with conventional technology. The reason is device 101 may not have access to IP network 107 without first authenticating with AUSF 103b and may have no ability to request a parent certificate from AUSF 103b that chains to (or can be verified through) root certificate Cert.CI-root 105a stored on device 101. Without (i) previously recording Cert.Provider 104c and (ii) verifying a received Cert.Provider 104c

through a root certificate Cert.CI-root 105a (or some other certificate previously securely stored in device 101 before connecting with MNO 102), then device 101 may not feasibly be able to verify a received Cert.Provider 104c and thus authenticate MNO 102 and network 103.

Device 101 can derive an ephemeral public key ePK.TLS.Device 101n and an associated secret key or private key eSK.TLS.Device 101q, using a random number, a key pair generation function, and the set of cryptographic parameters for a TLS session. Device 101 can utilize a random number generator similar such as hardware random number generator in order to derive private key eSK.TLS.Device 101q and input the random number into the key pair generation function. A key pair generation function could be included in application software running on device 101, and exemplary software for key pair generation algorithm can comprise OpenSSL, crypto++, or Mozilla libraries. The resulting ephemeral public key ePK.TLS.Device 101n can be sent to MNO 102 and network 103 in a message 218 below in Figure 2 in a TLS 1.3 “Client Hello” message.

Device 101 could use a set of cryptographic algorithms 101s in order to process and use ePK.TLS.Device 101n and eSK.TLS.Device 101q with TLS messages such as a “client hello”, a “server hello”, and subsequent TLS messages. Private key eSK.TLS.Device 101q and a public key for network 102 of ePK.TLS.Network 103n can be input into an ECDH key exchange algorithm within cryptographic algorithms 101sa in order to mutually derive a shared secret key. The shared secret key can be used to generate or derive a symmetric ciphering key and a MAC key. Device 101 could use the PKI keys with cryptographic algorithms 101s and cryptographic parameters 101x in order to mutually derive a symmetric ciphering key and a MAC key with network 102, as specified in IETF RFC 8446 titled “The Transport Layer Security (TLS) Protocol Version 1.3”, which is herein incorporated by reference in its entirety. Note that device 101 and network 103 could also use the PKI keys to mutually derive a symmetric ciphering key and a MAC key using previous versions of TLS as well, such as TLS version 1.2.

Cryptographic algorithms 101s can include the steps and logic for processor 101p in device 101 to conduct in order for device 101 to securely communicate with MNO 102 and network 103. Cryptographic algorithms 101s can include at least symmetric ciphering algorithms, a random number generator, a key pair generation algorithm, digital signature algorithms, asymmetric ciphering algorithms, and key exchange algorithms. Cryptographic algorithms 101s can also include a key verification step for verifying that a received public key is valid for an ECC curve. Cryptographic algorithms 101s can use libraries available from example cryptographic suites such as OpenSSL, crypto++, BouncyCastle, or Mozilla, and other possibilities exist as well without departing from the scope of the present disclosure. Cryptographic algorithms 101s can use inputs of keys such as public keys, private keys, and/or symmetric keys along with cryptographic parameters 101x in order to for device 101 to process cryptographic data including ciphertext, key exchanges, and digital signatures.

Cryptographic parameters 101x can specify values or settings for (i) conducting an ECDH or ECDHE key exchange, (ii) mutually deriving a shared secret, (iii) mutually deriving a symmetric ciphering key from the shared secret, (iv) using a symmetric ciphering algorithm with the symmetric ciphering key, and (v) using a digital signature algorithm. As contemplated herein, cryptographic parameters 101x may also be referred to as parameters 101x. Each of device 101, MNO 102, and network 103 and device provider 104 can record at least one compatible subset of parameters within a set of cryptographic parameters 101x. Parameters 101x can specify values for an elliptic curve cryptography (ECC) curve name, key length, key formatting (e.g. compressed or uncompressed), encoding rules, etc.

As contemplated herein, the parameters 101x and cryptographic algorithms 101s used with ECC PKI keys and a key exchange in the present invention can be compatible and substantially conform with ECC algorithms and keys as specified in (i) the IETF Request for Comments (RFC) 6090 titled “Fundamental Elliptic Curve Cryptography Algorithms”, and (ii) IETF RFC 5915 titled “Elliptic Curve Private Key Structure”, and also subsequent and related versions of these standards. Other possibilities exist as well for cryptographic parameters 101x without departing from the scope of the present invention.

For use of ECC algorithms, parameters 101x can specify elliptic curve names such as, but not limited to NIST P-256, sect283k1, sect283r1, sect409k1, sect409r1, and other possibilities exist as well. Further, elliptic curves that do not depend on curves currently specified by the National Institute of Standards and Technology (NIST) could be utilized as well, such as, but not limited to, Curve22519, curve448, or FourQ. Parameters 101x can specify domain parameters for nodes in system 100 to calculate values or numbers in a compatible manner, such as common base point G for use with ECC PKI key pairs and a defining equation for an elliptic curve.

Network 103 can comprise a collection of servers and also operate as a cloud service. As depicted for system 100 in Figure 1, network 103 can communicate with device 101, MNO 102, and device provider 104 through IP network 107. Network 103 can include a network identity of network ID 103a, which could comprise a domain name, a name, or a string to uniquely identify network 103 in a system 100. In exemplary embodiments, at least a portion of network ID 103a is included in the realm of device identity comprising a NAI for SUPI 101a. Network 103 can include at least one server 111 as depicted in Figure 1. Server 111 can include hardware components similar to those of a device 101 depicted in Figure 1, except generally with larger capacities appropriate for a server 111. Server 111 can also operate as a host computing environment with physical hardware for a virtual machine to operate as a guest computing environment. In an exemplary embodiment, the AUSF 103b can comprise a virtual machine operating on server 111.

A server 111 can include random access memory (RAM) 111a, storage memory 111b, at least one system bus 111c, and at least one network interface 111d. As within a server 111 operating in a

network 103, server 111 can include at least one processor 103p in order to store and record data as well as communicate with other nodes over an IP network 107, such as with MNO 102 and device provider 104 through an IP network 107. Processor 103p can also be referred to as a central processing unit (CPU) 103p. Processor 103p can comprise a general purpose processor appropriate for the computational requirements for a server 101, and may operate with multiple different processor cores, including field programmable gate arrays (FPGA).

A processor 103p can comprise exemplary ARM® based processors or an Intel® based processor such as belonging to the XEON® family of processors, and other possibilities exist as well. Processor 103p can utilize the system bus 111c to fetch instructions from RAM memory 111a or storage memory 111b within a server 111 and operate on the instruction. A processor 103p can include components such as registers, accumulators, and logic elements to add, subtract, multiply, and divide numerical values and record the results in memory. In exemplary embodiments, at least one processor 103p within server 111 can be used to conduct the steps and message flows depicted in Figure 2 below.

RAM 111a may comprise a random access memory for Server 111. RAM 111a can be a volatile memory providing rapid read/write memory access to processor 103p. RAM 111a could be located on a separate integrated circuit in server 111. The system bus 111c may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures including a data bus. System bus 111c connects components within server 111, such as transferring electrical signals between the components illustrated for a server 111. Server 111 can include multiple different versions of bus 111c to connect different components, including a first system bus 111c between CPU 103p and RAM 111a (which could be a memory bus), and a second system bus 111c between CPU 103p and network interface 111d, which could be a SPI bus, a PCI bus, or similar data busses.

Server 111 may also operate a network interface 111d, where network interface 111d can be used to connect and communicate with other nodes such as depicted in Figure 1 and also Figure 2 below. Network interface 111d can comprise a physical interface connected to system bus 111c for server 111. In exemplary embodiments, network interface 111d can comprise an Ethernet or fiber optic wired connection. Network interface 111d can connect server 111 to plurality of computing devices and other servers through networks such as the globally routable public Internet.

Nonvolatile memory 111b or “storage” 111b (which can also be referred to herein as “storage memory 111b”) within server 111 can comprise a non-volatile memory for long-term storage of data, including times when server 111 may be powered off. Storage memory 111b may be a NAND flash memory or a NOR flash memory and record firmware for server 111, such as a bootloader program and an operating system. Storage memory 111b can record long-term and non-volatile storage of data or files for server 111. In exemplary embodiments, the network identity 103a is recorded in storage

memory 111b when server 111 is powered off, and the network identity 103a along with a network database 103d are moved by CPU 103p into RAM 111a when server 111 powers on.

Storage memory 111b can operate as a traditional hard disk drive or a solid state drive (SSD), or as a storage area network (SAN). Storage memory 111b can also comprise a plurality of spinning hard disk drives in a redundant array of independent disks (RAID) configuration. Storage memory 111b may also be referred to as “server storage” and can include exemplary file systems of FAT16, FAT 32, NTFS, ext3, ext4, UDF, or similar file systems. As contemplated herein, the terms “memory 111b”, “storage memory 111b”, and “nonvolatile memory 111b” can be considered equivalent.

Network 103 can include an authentication server function AUSF 103b as specified in ETSI TS 133 501 V15.1.0, as well as subsequent or related versions of ETSI standards. In general, an authentication server function can communicate with a security anchor function SEAF 102b from MNO 102. AUSF 103b can receive a SUCI 101b for device 101 from SEAF 102b, where MNO 102 receives SUCI 101b from device 101 via a wireless network and radio 102r. Additional details for the communication and operation of AUSF 103b for network 103 is provided below in Figure 2. In exemplary embodiments, AUSF 103b and device 101 can support EAP-TLS authentication, where EAP-TLS authentication for device 101 using a wireless network is described in Annex B of ETSI TS 133 501 V15.1.0. AUSF 103b can send and receive messages with device 101 through MNO 102 and also send and receive messages with device provider 104 through IP network 107. Network 103 can include at least one processor 103p, where processor 103p can operate in a similar manner as a processor 102p described above for MNO 102. Processor 103p could be included in a server 111 and also within AUSF 103b. A network 103 can operate with a plurality of processors 103p and/or servers 111 in order to perform the steps and communicate the messages depicted and described for a network 103 in Figure 2 below.

Network 103 can derive an ephemeral public key ePK.TLS.Network 103n and an associated secret key or private key eSK.TLS.Network 103q, using a random number, a key pair generation function, and the set of cryptographic parameters for a TLS session. Network 103 can utilize a random number generator similar such as hardware random number generator in order to derive the private key eSK.TLS.Network 103q and input the random number into the key pair generation function. A key pair generation function could be included in application software running in network 103. The PKI keys depicted and associated cryptographic algorithms and cryptographic parameters could be operated within AUSF 103b, where AUSF 103b operates on a server 111. The cryptographic algorithms and cryptographic parameters operated within AUSF 103b could be equivalent to cryptographic algorithms 101s and cryptographic parameters 101x used by device 101. Exemplary software for key pair generation algorithms and cryptographic algorithms equivalent to cryptographic algorithms 101s within network 103 can comprise OpenSSL, crypto++, or Mozilla libraries.

The resulting ephemeral public key ePK.TLS.Network 103n can be sent to MNO 102 and device 101 in a message 227 below in Figure 2 in a TLS 1.3 “Server Hello” message as key share 103n. Network 103 could use a set of cryptographic algorithms in order to process and use ePK.TLS.Network 103n and eSK.TLS.Network 103q with TLS messages such as a “client hello”, a “server hello”, and subsequent TLS messages. Private key eSK.TLS.Network 103q and a public key for device 101 of ePK.TLS.Device 101n can be input into an ECDH key exchange algorithm in order to mutually derive a symmetric ciphering key and a MAC key. AUSF 103b could use the PKI keys with cryptographic algorithms and cryptographic parameters in order to mutually derive a shared secret key and then a symmetric ciphering key and a MAC key with device 101, as specified in IETF RFC 8446 titled “The Transport Layer Security (TLS) Protocol Version 1.3”, which is herein incorporated by reference in its entirety. Note that device 101 and network 103 could also use the PKI keys to mutually derive a symmetric ciphering key and a MAC key using previous versions of TLS as well, such as TLS version 1.2.

As depicted in Figure 1, a network 103 can also operate a network database 103d. A network database 103d could comprise a collection of servers and storage memory for storing values associated with a plurality of devices 101 in order to support the operation of a system 100. A network database 103d could store values, strings, or numbers for device 101 such as a SUCI 101b received from MNO 102 and a device provider certificate 104c for device 101 received from device provider 104, and other data could be stored in a network database 103d as well. Exemplary data for an exemplary network database 103d is depicted and describe in connection with Figure 3 below. Data within a network database 103d could be recorded or stored as network 103 receives data for a device 101 from a MNO 102 and a device provider 104. Subsequent communications after receipt of the data could use a network database 103d in order to select stored values for communication with device 101.

Device provider 104 can comprise an entity or set of servers for authenticating or managing plurality of devices 101. Both device provider 104 and MNO 102 could operate a plurality of server 111. Although depicted in Figure 1 and Figure 2 as “device provider 104”, an entity such as a device owner, a device manufacturer, or a device distributor could record the data and perform the functions of a “device provider 104”. In general, a device provider 104 can comprise an entity or collection of servers which is the source for providing Cert.Provider 104c to device 101 before device 101 connects with MNO 102, where Cert.Provider 104c can be used by device 101 to authenticate MNO 102 and with AUSF 103b below in Figure 2. As contemplated herein, a “device provider” may also be referred to as a “provider”. Consequently, a device provider secret key SK.Provider 104e can be referred to as a “provider secret key 104e”, a device provider certificate Cert.Provider 104c can be referred to as a “provider certificate 104c”, etc.

Although depicted as “device provider” in Figure 1 and Figure 2, the device provider 104 could be a device owner, a device distributor, or a device manufacturer. In general, a device provider has a

sufficient level of ownership or control of device 101 before device 101 connects with MNO 102 such that device provider 104 could record or store the device identity SUPI 101a, and at least a certificate 101c for device 101 before device 101 connects with MNO 102. For some embodiments, device provider 104 could also store a device provider certificate 104c before device 101 connects with MNO 102. For some embodiments, device provider 104 could also be a device user, where the device user operates a plurality of different devices 101. As one example, device provider 104 could be a large enterprise responsible for hundreds or more devices 101 that are geographically dispersed, and the enterprise may prefer to support the authentication and networking connectivity for device 101 using the steps and messages depicted for a system 200 in Figure 2 below.

Device provider 104 can include a provider identity of provider ID 104a. Provider ID 104a can comprise a name or unique identifier, including a domain name or at least a portion of a realm in a NAI, in order to uniquely identify device provider 104 in a system 100, since a system 100 could include a plurality of different device providers 104. At least a portion of the provider ID 104a can be included in the SUPI 101a and SUCI 101b, such as within the realm portion of the SUPI 101a and SUCI 101b, as depicted in Figure 1. Although a single provider ID 104a is depicted in Figure 1, a device provider could use a plurality of different related provider ID 104a names or identities, such as “company-1”, “company-2”, etc. The use of different provider ID 104a names is also depicted and described for in a device database 104d in Figure 3 below.

Different devices 101 could include the different provider ID 104a names in the SUCI 101b, such as a first set of devices using an exemplary value of “company-1” in the realm portion of a SUCI 101b and a second set of devices using an exemplary value of “company-2” in the realm portion of SUCI 101b. In exemplary embodiments, the provider ID 104a in the realm portion of a SUCI 101b can be used to select from device database 104d the SK-ECIES.Provider 104b for an ECIES decryption function 104j, such that different devices 101 can use different corresponding PK-ECIES.Provider 101e for a SUPI/ECIES encryption function 101j by the different devices. The use of different provider ID 104a names and different secret keys SK-ECIES.Provider 104b for device provider 104 is also depicted for a device database 104d in Figure 3 below.

Device provider 104 can store a plurality of device identities comprising SUPI 101a for a plurality of devices 101. Although depicted as external to device database 104d in Figure 1, the values of SUPI 101a could be recorded within device database 104d along with the unique user name portion for each of the devices 101. Note that for a given device 101 in a system 100, a user name may not be globally unique, but the combination of user name and realm for a SUPI 101a can be globally unique in preferred embodiments. Device provider 104 can also store the concealed identity SUCI 101b received from a network 103 and originated by device 101 in order to conduct an ECIES 104j decryption step to convert the SUCI 101b into a SUPI 101a for a device 101.

Device provider 104 can also store or record a plurality of device certificates 101c corresponding to a plurality of devices in a system 100. The device provider 104 could be responsible for initial configuration of devices 101 or receive initial configuration data for devices 101, where the initial configuration of device 101 can include the generation of a device certificate 101c, and device provider 104 could store the device certificate 101c for each of the devices 101. Although depicted as external to a device database 101d, device provider 104 could store a plurality of device certificates 101c in a device database 101d. By device provider 104 storing a device certificate 101c and providing the device certificate 101c to a network 103, the authentication by network 103 of device 101 can be simplified, as described below in Figure 2 (since network 103 could trust device provider 104 and receive the device certificate 101c from device provider 104, where network 103 may not necessarily trust device 101 when a certificate 101c is received from device 101).

Device provider 104 can store or record a secret key for the device provider 104 of SK-ECIES.Provider 104b in order to conduct ECIES 104j decryption of the user name in a received SUCI 101b. The private key can be associated with a set of cryptographic parameters ECIES Parameters 101f for using the private key (such as an ECC curve name), and the cryptographic parameters could be used with ECIES 104j decryption function. Note that a device provider 104 could use (i) a first SK-ECIES.Provider 104b with a first provider ID 104a with ECIES 104j decryption function and (ii) a second SK-ECIES.Provider 104b with a second provider ID 104a with ECIES 104j decryption function, which is also depicted within a device database 104d in Figure 3 below.

ECIES Parameters 101f can specify a named curve associated with private key SK-ECIES.Provider 104b. As described above, the device 101 can record a public key corresponding to SK-ECIES.Provider 104b, where the public key for device 101 can comprise PK-ECIES.Provider 101e as depicted in Figure 1. The value of PK-ECIES.Provider 101e could be generated by device provider 104 using SK-ECIES.Provider 104b and ECIES Parameters 101f. PK-ECIES.Provider 101e could be provided to device 101 before device 101 sends a SUCI 101b to MNO 102, such as during device manufacturing or device distribution or device configuration. Different values of SK-ECIES.Provider 104b and corresponding ECIES Parameters 101f can be selected by device provider 104 using (i) at least a portion of provider ID 104a in a realm within a received SUCI 101b for device 101 and (ii) a device database 104d. SK-ECIES.Provider 104b can comprise a first private key for named curve P-256, a second public key for named curve 25519, where the selection of the first private key or second private key could be based on ECIES Parameters 101f, which could be associated with the provider ID 104a in a realm received in a SUCI 101b for device 101. Note that the transmission of a SUCI 101b by device 101 can also include a specified set of ECIES parameters 101f in order to identify a named curve or parameters associated with ePK-ECIES.Device 101h transmitted by device 101 along with the SUCI 101b.

Device provider 104 can include a decryption function ECIES 104j in order to convert a SUCI 101b into a SUPI 101a. The decryption function ECIES 104j can comprise the identity decryption scheme as described in Annex C: “Protection schemes for concealing the subscription permanent identifier” in ETSI standard TS 33.501 version 15.1.0. In summary, device provider can conduct an elliptic curve Diffie-Hellman key exchange using (i) private key SK-ECIES.Provider 104b, (ii) a received device public key ePK-ECIES.Device 101h received along with SUCI 101b, and (iii) parameters ECIES Parameters 101f in order to generate an encryption key and a message authentication code (MAC) key. The user portion of SUCI 101b, or the encrypted user name, can be input into the decryption function ECIES 104j along with the derived encryption key and MAC key in order to generate SUPI 101a which contains the user name as a plaintext. A MAC code generated by ECIES 104j using the MAC key can be used to verify message integrity for the SUCI 101b by device provider 104. The realm portion of SUCI 101b can be received as plaintext and may not be altered or changed by ECIES 104j. In exemplary embodiments, a MAC code output by ECIES 104j is calculated over the entire SUCI 101b, including an encrypted user name for device 101, and thus the integrity or authentication of a received plaintext realm within the SUCI 101b can be verified by comparing the received MAC code for a SUCI 101b and a calculated MAC code output the ECIES 104j.

A certificate for device provider 104 can comprise Cert.Provider 104c and can include the device provider identity comprising provider ID 104a. A certificate 104c for device provider 104 can include a public key corresponding to the device provider private key SK.Provider 104e, a set of cryptographic parameters for the device provider public key, the identity of a certificate authority, and a digital signature for the certificate authority, as well as a specified digital signature algorithm used for the digital signature. A certificate 104c for device provider 104 and other nodes in a system 100 can comprise a certificate supporting X.509 v3 formats, and other formats or structures for a device provider certificate Cert.Provider 104c and other certificates as contemplated herein are possible as well. The cryptographic parameters can specify parameters for both the device provider public key and also the digital signature from a certificate authority. Cryptographic parameters in a certificate can specify algorithms such as RSA or elliptic curve cryptography (ECC) values, ECC curve names, key lengths in bits, signature algorithms, validity dates, encoding rules, etc. The signature of the certificate authority or certificate issuer for Cert.Provider 104c could be included in device provider certificate 104c.

Note that a device provider 104 could use a plurality of different device provider certificates 104c, and the device provider certificate 104c for a device 101 could be selected from a device database 104d for a device 101 using the SUPI 101a output from decryption function ECIES 104j. In some exemplary embodiments, the device provider certificate Cert.Provider 104c is also recorded in a device 101 as depicted in Figure 1 before device 101 sends a SUCI 101b to MNO 102, such as during a device configuration step or a device manufacturing step. In other exemplary embodiments, device 101 may not record Cert.Provider 104c before device 101 sends a SUCI 101b to MNO 102, and for these

embodiments Cert.Provider 104c could be received and verified by device 101 in a TLS handshake, where device 101 could verify Cert.Provider 104c using another “parent” certificate for Cert.Provider 104c such as certificate issuer 105 root certificate Cert.CI-root 105a.

A device provider 104 can record and use a device provider private key SK.Provider 104e (e.g. “secret key”). The private key can correspond to the public key in the device provider certificate Cert.Provider 104c. In exemplary embodiments, device provider 104 (i) internally generates a random number for the device provider private key SK.Provider 104e and using cryptographic algorithms 104s, (ii) generates the device provider public key in a Cert.Provider 104c, and (iii) stores a copy of a set of cryptographic parameters within the certificate Cert.Provider 104c, such as the values for an elliptic curve named curve and specification of digital signature algorithms. The generation of the public key for device provider certificate Cert.Provider 104c could be before or during device 101 manufacturing or distribution and before device 101 sends a SUCI 101b to a MNO 102. In this manner, a device provider certificate Cert.Provider 104c could be written to device 101 before device 101 sends a SUCI 101b to MNO 102.

In some exemplary embodiments, device provider 104 operates a parent or “intermediate” certificate authority or certificate issuer in order to generate and sign Cert.Provider 104c. The public key for SK.Provider 104e in Cert.Provider 104c could be signed by a device provider secret key SK.Provider-CI 104g for Cert.Provider-CI 104f. The public key for Cert.Provider-CI 104f could be signed by SK.Root 105b from certificate issuer 105, where the certificate authority/issuer signature in Cert.Provider-CI 104f can be verified with root certificate Cert.CI-root 105a which can be stored in a device 101. Although a single level of a certificate issuer between device provider certificate Cert.Provider 104c and root certificate Cert.CI-root 105a is depicted in Figure 1, a device provider 104 could operate multiple levels of intermediate certificates between device provider certificate Cert.Provider 104c and root certificate Cert.CI-root 105a. In addition, although a single root certificate issuer 105 is depicted in Figure 1, a system 100 could include multiple root certificate issuers 105, and device provider 104 could record multiple certificates Cert.Provider-CI 104f or Cert.Provider 104c signed by different root certificate issuers 105 with different SK.Root 105b from the different certificate issuers 105.

Cryptographic algorithms 104s can include the steps and logic for processor 104p in device provider 104 to conduct in order for device provider 104 to securely communicate with network 103. Note that a processor 104p can be equivalent to a processor 103p for a server 111 above, and a device provider 104 can operate a plurality of servers 111. Cryptographic algorithms can include symmetric ciphering algorithms, a random number generator, a key pair generation algorithm, digital signature algorithms, asymmetric ciphering algorithms, and key exchange algorithms. Cryptographic algorithms can also include a key verification step for verifying that a received public key is valid for an ECC curve. Cryptographic algorithms 104s can use libraries available from example cryptographic suites

such as OpenSSL, crypto++, BouncyCastle, or Mozilla, and other possibilities exist as well without departing from the scope of the present invention. Cryptographic algorithms 104s can use inputs of keys such as public keys, private keys, and/or symmetric keys along with cryptographic parameters 104x in order to for device provider 104 to process cryptographic data including ciphertext, key exchanges, and digital signatures.

Cryptographic parameters 104x can specify values or settings for (i) conducting an ECDH or ECDHE key exchange to obtain or derive a shared secret, (ii) mutually deriving a symmetric ciphering key using at least the derived shared secret, (iii) using a symmetric ciphering algorithm with the symmetric ciphering key, and (iv) using a digital signature algorithm. Cryptographic algorithms 104s for device provider 104 can correspond or be equivalent to cryptographic algorithms 101s for device 101, and cryptographic parameters 104x for device provider 104 can correspond or be equivalent to cryptographic parameters 101x for device 101. Note that a server 111 for a network 103 or device provider 104 can store and operate with cryptographic parameters equivalent to the depicted cryptographic parameters 104x.

Device provider 104 can operate a plurality of servers with at least one processor 104p. A processor 104p can comprise a central processing unit (CPU) or a “system on a chip” and be similar to a processor 102p for a MNO 102 described above. Device provider user interface 104u can comprise a web-based interface for administrators of device provider 104 to manage the function and operation of device provider 104 as well as manage a plurality of devices 101. User interface 104u can support the addition of new devices 101 in a system 100, the deletion or revocation of existing devices 101 in a system 100, etc. User interface 104u can also allow changes in values and tables for a plurality of devices 101 in a device database 104d, where a device database 104d is depicted and described in connection with Figure 3 below.

Device database 104d can record a plurality of values for a plurality of devices 101 in order for device provider 104 to manage or operate communications with device 101 through a network 103. Device database 104d can record for each device exemplary values of SUPI 101a, device certificate 101c, the provider ID 104a used by device 101, a device provider certificate 104c used by a device 101, etc. Additional details regarding the data stored in a device database 104d are depicted and described in connection with Figure 3 below.

Although Figure 1 depicts MNO 102, network 103, and device provider 104 as separate nodes or entities, in some exemplary embodiments different entities or function of the nodes could be combined. In one embodiment, a MNO 102 and a network 103 could be combined, such that a MNO 102 controls or operates the network 103. In this embodiment, then a MNO 102 would operate both a SEAF 102b and an AUSF 103b. Or, a network 103 could control or operate a MNO 102, such that the network 103 controls or operates both a SEAF 102b and an AUSF 103b. In addition, although a MNO 102 is depicted in Figure 1, the wireless networking technology operated by MNO 102 with radios 102r

could use WiFi based technology (e.g. based on 802.11 standards) and a collection of geographically distributed WiFi access points. The WiFi access points could be controlled by network 103 and a security anchor function 102b could be omitted. In another embodiment, the function and operation of a network 103 and a device provider 104 could combined, such that either (i) network 103 records the data and operates the function of a device provider 104, or (ii) a device provider 104 operates an AUSF 103b. Other possibilities exist as well for the combination of the data and functions for a MNO 102, network 103, and device provider 104 without departing from the scope of the present invention.

Figure 2

Figure 2 is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a mobile network operator, a network, and a device provider, in accordance with exemplary embodiments. System 200 can include a device 101, a mobile network operator (MNO) 102, a network 103, and a device provider 104. The nodes or entities within system 200 were also depicted and described in connection with Figure 1 above, where Figure 2 depicts exemplary steps for the nodes and exemplary message flows between the nodes. Although a single device 101, MNO 102, network 103, and device provider 104 are depicted in a system 200, a system 200 could include a plurality of each of the depicted nodes connected via an IP network 107. In addition, data recorded for device 101 and device provider 104 depicted and described above in Figure 1 (other than ephemeral PKI keys) can be recorded in the device 101 and device provider 104 depicted in Figure 2 before the steps and message flows are conducted in Figure 2.

In order to support authentication of devices 101 with network 103, a MNO 102 could establish a secure session 201 with network 103. Secure session 201 could comprise establishing a secure communications link between the two servers using protocols such as TLS, IPSec, a virtual private network (VPN), a secure shell (SSH), or similar networking, transport, or application layer technologies in order to establish secure communications between MNO 102 and network 103. Secure session 201 can utilize certificates for each of MNO 102 and network 103 in order to provide mutual authentication and mutual key derivation for a symmetric encryption key in secure session 201. Secure session 201 can also be conducted over IP network 107, although the secure session 201 could also be established or conducted through a private network. Other possibilities exist as well for establishing a secure session 201 between MNO 102 and network 103 without departing from the scope of the present disclosure.

In order to support authentication of devices 101 with network 103, a network 103 and device provider 104 could establish a secure session 202. Secure session 202 can be equivalent to secure session 201 described in the paragraph above. Secure session 202 could use certificates from both network 103 and device provider 104 for mutual authentication and the secure derivation of encryption keys for the secure session. Note that subsequent data transmitted from device provider 104 to network

103 through the secure session 202 can be trusted by network 103 on the basis of the trust for secure session 202. For example, network 103 and AUSF 103b could receive a device certificate 101c for a device 101, where the device certificate 101c may not be authenticated with a certificate chain and a root certificate recorded or used by network 103. However, network 103 could still trust the device certificate 101c since the device certificate 101c could be received through secure session 202, where secure session 202 can be mutually authenticated using certificates for both network 103 and device provider 104. In some exemplary embodiments, both secure session 201 and secure session 202 are established using TLS 1.3.

At step 203, device 101 can power on and load firmware and configuration data in order to connect with wireless network 102. A step 203 can also include firmware or software in device 101 loading a RF module or radio 101r within device with an operating configuration such as MNO parameters 101g in Figure 1 (such as frequencies to use, wireless technology to use such as 5G or WiFi 6, etc.). Step 203 can include device 101 scanning for available mobile networks 102 and then selecting a preferred mobile network. In exemplary embodiments, device 101 scans for mobile network identities broadcast in system information blocks, where the mobile network identity such as mobile network ID 102a matches the value for a mobile network in a realm for a SUPI 101a recorded by device 101.

In other words, device 101 could include multiple SUPI 101a with different realms for different networks 103 and device providers 104, and search for a MNO 102 in an RF beacon or signal that matches a realm for one of the multiple SUPI 101a. For other embodiments, the MNO ID 102a may not be included in a portion of a realm within a SUPI 101a, and a device 101 could select the MNO 102 using MNO parameters 101g. At the conclusion of a step 203, device 101 can select and store a SUPI 101a to use with a MNO 102. A step 203 can also include device 101 selecting MNO parameters 101g based on an observed and available MNO 102 for a radio 101r in device 101. The selection of a SUPI 101a for device 101 in a step 203 can include the selection of (i) a device provider 104 provider ID 104a for a realm of the SUPI 101a and (ii) a selection of network identity network ID 103a for a realm of SUPI 101a. The values for PK-ECIES.Provider 101e and ECIES Parameters 101f can be selected in a step 203 based on the provider ID 104a for the SUPI 101a. As one example a SUPI 101a selected in a step 203 could comprise an exemplary value of “XXXXXX@company.aws.com”, where XXXXXX is the user name, “company” is the provider ID 104a and “aws.com” is the network ID 103a.

At step 204, device 101 can conduct a radio resource connection (RRC) request with MNO 102 using a radio 101r and the selected MNO parameters 101g. The RRC can transition a radio 101r in device 101 from an idle state to an active state. In the active state, device 101 can transmit messages or data to MNO 102. A step 204 can also include device 101 generating an ephemeral PKI key pair of ePK.ECIES.Device 101h and eSK.ECIES.Device 101i. At step 101j, device 101 can conduct the ECIES 101j encryption function with secret key eSK.ECIES.Device101i and public key PK-ECIES.Provider 101e in order to derive a symmetric ciphering key and an MAC key. The user name

portion of a SUPI 101a can be input into the ECIES 101j along with the symmetric key and the MAC key in order to output a first portion the SUCI 101b comprising an encrypted user name for the device 101.

The realm portion of a SUPI 101a can be input into the ECIES 101j along with the MAC key in order to output a MAC code covering the realm portion the SUCI 101b, such that the realm portion can be verified using the MAC code. Or, a single MAC code for both the first portion of the SUCI 101b and the realm portion of the SUCI 101b could be generated or calculated in a step 204. As depicted in Figure 2, the combination of an encrypted user name, selected provider ID 104a for a realm, and the selected network ID 103a for a realm can comprise a value 204a. The value 204a can also include a MAC code using the MAC key from ECIES 101j encryption function, such that a device provider 104 receiving and decrypting the value 204a comprising at least the SUCI 101b and the MAC code can verify the integrity of the value 204a.

Device 101 can then send the selected MNO 102 through (i) a wireless network operated by MNO 102 and (ii) the radio resource connection from step 203 a message 205, where the message 205 includes the public key ePK.ECIES.Device 101h and the value 204a. MNO 102 and SEAF 102b can receive the message 205 and conduct a step 206 in order to process the message 205. At step 206, SEAF 102b can (i) read the plaintext realm for SUCI 101b in value 204a, and (ii) select the network 103 using the network ID 103a in a realm for SUCI 101b. In a preferred exemplary embodiment as depicted in Figure 2, the suffix of the realm can comprise the network ID 103a, but in other embodiments the prefix of the realm could comprise the network ID 103a.

Note that MNO 102 and network 103 could have prior commercial and technical agreements for MNO 102 to forward device 101 authentication data to network 103 and AUSF 103b. If MNO 102 has no commercial relationship with network 103, or secure session 201 could not be established, or MNO 102 cannot forward the data in message 205, then MNO 102 can send device 101 an error message with an error code explaining the failure. Device 101 could then take corrective steps, such as selecting a different MNO 102 or different network 103 via a different SUPI 101a for device 101 in order to send a message 205 to the different MNO 102.

MNO 102 can then send network 103 and AUSF 103b a message 207 through secure session 201, where (i) message 207 includes ePK.ECIES.Device 101h and the value 204a, and (ii) the value 204a includes SUCI 101b. The AUSF 103b for network 103 can receive the message 207. At step 208, the AUSF 103b can read and process the plaintext data in the realm for SUCI 101b in value 204a. At step 208, the AUSF 103b can select the device provider 104 using the device provider ID 104a in the realm for SUCI 101b. In a preferred exemplary embodiment as depicted in Figure 2, the prefix of the realm can comprise the provider ID 104a.

Note that network 103 and device provider 104 could have prior commercial and technical agreements for network 103 to forward device 101 authentication data to device provider 104. If

network 103 has no commercial relationship with device provider 104, or secure session 202 could not be established, or network 103 cannot forward the data in message 207, then network 103 can send MNO 102 an error message with an error code explaining the failure. MNO 102 could then send device 101 an error code explaining the failure, such that device 101 could take corrective steps, such as selecting a different MNO 102 or different network 103 in order to send a message 205 to the different MNO 102. A step 208 can include AUSF 103b storing SUCI 101b in a network database 103d for subsequent use in identifying device 101, such as when receiving a “client hello” message 218 below with a SUCI 101b added to message 218 by SEAF 102b.

Network 103 and AUSF 103b can then send a message 209 to device provider 104 through secure session 202, where message 209 can include (i) MNO ID 102a, and (ii) ePK.ECIES.Device 101h and the value 204a, where the value 204a includes SUCI 101b. Note that the inclusion of MNO ID 102a can be useful for device provider 104 in a message 209, since (i) device 101 could connect with multiple different MNO 102, and (ii) network 103 could support multiple different MNO 102, and (iii) device provider 104 may need to know which MNO 102 is used by device 101 for billing or administrative purposes. In other words, without including MNO ID 102a in message 209 then device provider 104 may not know which MNO 102 that device 101 attempts to connect with.

Device provider 104 can then conduct a series of steps in order to process the message 209. At step 210, device provider 104 can read the plaintext provider ID 104a in SUCI 101b and use device database 104d in order to select (i) secret key SK-ECIES.Provider 104b and (ii) ECIES Parameters 101f. As depicted in Figure 3 below for a device database 104d, a provider ID 104a can be uniquely associated with secret key SK-ECIES.Provider 104b and (ii) ECIES Parameters 101f. Device provider 104 can then conduct a verification of the received ePK.ECIES.Device 101h using the ECIES Parameters 101f. The verification step can comprise verifying that the received public key is on the curve specified by ECIES Parameters 101f, the public key value does not comprise the “point at infinity”, etc.

Device provider 104 can then conduct a step 104j' using ECIES decryption function 104j in order to convert the ciphertext user name in SUCI 101b into plaintext. ECIES 104j was depicted and described in connection with Figure 1 above. A step 104j' can also include deriving a MAC key and verifying a MAC code within value 204a in order to check the message integrity of SUCI 101b. The check of message integrity of SUCI 101b in a step 104j' can include verifying the message integrity of a plaintext realm portion of the SUCI 101b using a MAC key from ECIES 104j. In other words, device provider can securely store a SUPI 101a for device 101 after step 104j'. Device provider 104 can conduct a step 211 with the plaintext user name to query a device database 104d for data pertaining to device 101, and a step 211 can include verifying that device 101 with the SUPI 101a is authorized to connect with MNO 102 and/or network 103. In a step 211, device provider 104 can determine the MNO 102 that device 101 attempts to connect with based on the MNO ID 102a received in a message 209 above. If device 101 is not authorized to connect with MNO 102, then device provider 104 could send

an error message to network 103 (for forwarding to device 101) and not proceed to the additional steps depicted in Figure 2.

Although an ECIES decryption function 104j is depicted in Figure 2 as being conducted by a device provider 104, in some exemplary embodiments a network 103 could conduct the ECIES decryption function 104j and send device provider 104 a message 209 with the SUPI 101a for a device 101 (instead of message 209 including a SUCI 101b). For these embodiments, then network 103 could store the values for secret key SK-ECIES.Provider 104b and parameters ECIES Parameters 101f, as well as operate the ECIES decryption function 104j. In other words, although Figure 1 depicts values for secret key SK-ECIES.Provider 104b and parameters ECIES Parameters 101f, as well as the ECIES decryption function 104j as within a device provider 104, a network 103 could store the data and perform a step 104j' in some exemplary embodiments. In some exemplary embodiments, network 103 could conduct the step 104j' (instead of device provider 104 as depicted in Figure 2) to decrypt the encrypted user portion within a SUCI 101b received from MNO 102 in a message 207 in order to generate a SUPI 101a for device 101. For the embodiment described in this paragraph, then a message 209 could include a SUPI 101a instead of a SUCI 101b, and network 103 could conduct the steps of 211 and 104j' instead of a device provider 104. For the embodiment described in this paragraph, then the depicted key of "PK-ECIES.Provider 101e" stored by device 101 can be associated with a network 103 instead of a device provider 104.

At step 212, device provider 104 can use the user name from SUPI 101a to query device database 104d for the device certificate cert.device 101c and also the device provider certificate cert.provider 104c recorded by device 101, as depicted and described above in connection with Figure 1. Note that a device certificate cert.device 101c can include either (i) the SUPI 101a as the device identity within the certificate or (ii) the user name portion from a NAI for device 101. For some exemplary embodiments, a step 212 could omit the selection of either device certificate 101c or device provider certificate 104c, and either certificate could be omitted in a subsequent message such as message 213.

A step 212 can also include device provider 104 collecting an OSCP stapling 212a through root certificate Cert.CI-root 105a. OSCP stapling is described in the TLS 1.3 RFC 8446, and OSCP stapling 212a provides an updated signature from parent certificate issuers that the certificate has not been revoked. In exemplary embodiments, OSCP stapling 212a is provided for cert.provider 104c, such that device 101 could verify the OSCP stapling 212a using the certificate chain for cert.provider 104c recorded by device 101. OSCP stapling 212a can be important for device 101 because device 101 may not have connectivity to IP network 107 without successfully conducting an EAP-TLS authentication with MNO 102, but connectivity to IP network 107 would be required to check certificate revocation lists without OSCP stapling 212a. In other words, without OSCP stapling 212a from a step 212, device 101 may not feasibly be able to check for certificate revocation for the device provider certificate

cert.provider 104c selected by device provider 104 in a step 212. Device 101 can receive cert.provider 104c in a subsequent “server hello” message in Figure 2.

Device provider 104 can then send network 103 and AUSF 103b a message 213 through secure session 202, where message 213 can include the selected device provider certificate Cert.Provider 104c from a step 212, device certificate Cert.Device 101c (possibly w/ SUPI 101a) from a step 212, SUCI 101b, and OSCP stapling 212a for device provider certificate 104c. Note that the inclusion of SUCI 101b with message 213 (and SUPI 101a in a Cert.Device 101c) can provide AUSF 103b the mapping between SUCI 101b and SUPI 101a for a device 101. Network 103 and AUSF 103b can receive the message 213 and conduct a step 214 to store (i) device provider certificate Cert.Provider 104c and (ii) device certificate Cert.Device 101c for the SUPI 101a and (iii) SUCI 101b used by device 101 in a network database 103d as depicted in Figure 3 below. In exemplary embodiments Cert.Provider 104c includes a subset of cryptographic parameters 104x and Cert.Device 101c includes a subset of cryptographic parameters 101x.

Network 103 could operate a network database 103d similar to device database 104d, where an exemplary network database 103d is depicted in Figure 3 below. Network 103 could store received certificates such as device provider certificate Cert.Provider 104c and device certificate Cert.Device 101c for a plurality of devices 101 along with the SUCI 101b for devices 101 received in messages 207 and SUPI 101a for devices 101 received in messages 213. AUSF 103b can determine the SUPI 101a for device 101 based on the SUPI 101a within a device certificate 101c, and both the values for SUPI 101a and string or file for device certificate 101c could be stored in a network database 103d as depicted in Figure 2 below. Or, in another exemplary embodiment the SUPI 101a could also be sent external to a device certificate in a message 213. Network 103 and AUSF 103b can then use a network database 103d to select device provider certificate Cert.Provider 104c and device certificate Cert.Device 101c in subsequent steps, such as a step 219a using the SUPI 101a or SUCI 101b for a device 101. In addition, a message 213 from device provider 104 to network 103 could include a user name or identity of device 101 separate from or instead of a SUPI 101a. In other words, the realm portion of a SUPI 101a is not required for a message 213 and a device identity in the form of a user name from a NAI for device 101 can be sufficient for network 103 to conduct subsequent steps and messages as depicted in Figure 2.

In addition, although message 213 with Cert.Provider 104c and Cert.Device 101c is depicted as being sent by device provider 104 after the receipt of a message 209, device provider 104 could send Cert.Provider 104c and Cert.Device 101c to network 103 before a message 209 and after the establishment of secure session 202. Device provider 104 could send Cert.Provider 104c and Cert.Device 101c for a plurality of different devices 101. By sending message 213 as depicted in Figure 2, then OSCP stapling 212a can be more current or more “fresh” for device 101. In addition, a network 103 could record a plurality of device provider certificates Cert.Provider 104c in a network database 103d before network 103 receives a message 213, and in this embodiment then a message 213 could

include an identifier of a device provider certificate Cert.Provider 104c (such as a certificate identity) instead of the full certificate as a *.pem file or equivalent. Further, a network 103 could record a plurality of device certificates Cert.Device 101c in a network database 103d before network 103 receives a message 213, and in this embodiment then a message 213 could include an identifier of a device certificate Cert.Device 101c (such as a certificate identity) instead of the full certificate as a *.pem file or equivalent.

Further, although device provider certificate Cert.Provider 104c and device certificate Cert.Device 101c are depicted in Figure 2 as being sent in the same message 213, a message 213 could be sent in parts, such as the data being transmitted via a physical interface in device provider 104 separately. A device provider certificate Cert.Provider 104c or device certificate Cert.Device 101c could be selected by device provider at a later point in time than depicted in Figure 2, such as device provider 104 sending device provider certificate Cert.Provider 104c and/or device certificate Cert.Device 101c after the receipt of a message 220 from network 103. A step 211 to select the device provider certificate Cert.Provider 104c or device certificate Cert.Device 101c could be conducted after device provider 104 receives a message 220. Other possibilities for the sequence and timing for a device provider 104 to send a network 103 a device provider certificate Cert.Provider 104c or device certificate Cert.Device 101c are possible as well without departing from the scope of the present disclosure.

Note that network 103 and AUSF 103b can trust Cert.Provider 104c and Cert.Device 101c, since the certificates can be delivered through the secure session 202. Or, in a step 214 network 103 could conduct a certificate verification of Cert.Provider 104c and Cert.Device 101c through a root certificate, such as Cert.CI-root 105a. Note that Cert.Provider 104c and Cert.Device 101c do not have to be verified through a root certificate stored by network 103, since the certificates are received through secure session 202, where secure session 202 can be established with other certificates for device provider 104 (and the other certificates for a secure session 202 could be verified through a different root certificate than the Cert.CI-root 105a recorded by device 101 as depicted in Figure 1). A step 214 can also include AUSF 103b verifying that AUSF 103b supports the cryptographic parameters 101x and 104x in the certificates received in a message 213. In other words, AUSF 103b can use a step 214 to verify that AUSF 103b can conduct a TLS 1.3 session using the parameters and certificates for Cert.Provider 104c and Cert.Device 101c received in a message 213.

At step 103q' AUSF 103b can generate an ephemeral private key for a TLS session with device 101 comprising eSK.TLS.Network 103q and the parameters within Cert.Provider 104c. In other words, AUSF 103b can read the named elliptic curve and key length from Cert.Provider 104c received in message 213 and generate a private key for the named curve and key length in a step 103q'. The use and function of an ephemeral secret key eSK.TLS.Network 103q was also described above in Figure 1. AUSF 103b can generate the corresponding public key ePK.TLS.Network 103n. A step 103q' could also be conducted after receiving a "client hello" message 218. AUSF 103b can then send device 101

an “EAP-TLS Start” message 215, which is also depicted as step 6 in Figure B.2.1.1-1 of 3GPP TS 33.501 version 15.1.0. Note that message 215 is sent to MNO 102 and MNO 102 can transmit message 215 through the radio resource connection established by device 101 in step 204 above. Where MNO 102 functions as a “pass through” entity for messages between network 103 and device 101, then the vertical line for MNO 102 in Figure 2 is omitted.

Device 101 can receive the “EAP-TLS” start message 215 and can conduct a series of steps in order to generate the “client hello” message. Although TLS version 1.3 according to IETF RFC 8446 is depicted in Figure 2, subsequent and related version of a transport layer security protocol could be used as well between device 101 and an AUSF 103b operating in network 103. In a step 216, device 101 using processor 101p can select parameters for a subsequent “client hello” message 218 using the cryptographic parameters in the stored device provider certificate 104c associated with a device provider ID 104a used in the realm of a SUCI 101b transmitted by device 101 in a message 205. In this manner, the set of cryptographic parameters for a “client hello” message 218 and a “server hello” message 227 can be reasonably assured to be compatible before the “client hello” message 218 is transmitted by device 101. In other words, in exemplary embodiments, a device 101 can store a device provider certificate 104c before device 101 sends a message 205, including storing the device provider certificate 104c during a configuration of device 101 or a secure session for device 101 before device 101 sends a message 205.

The selected parameters in a step 216 can include at least a digital signature algorithm such as the elliptic curve digital signature algorithm as supported by TLS version 1.3 (or another TLS version for messages 218 and 227) and also a named elliptic curve supported by the TLS version in subsequent messages 218 and 227. In exemplary embodiments, the set of cryptographic parameters selected from stored device provider certificate Cert.Provider 104c can be a subset of cryptographic parameters 101x for device 101. Or, in some exemplary embodiments device 101 could select the entire set or a larger subset of cryptographic parameters 101x in a step 216 in order to generate a “client hello” message 218. The selected set of cryptographic parameters for device 101 in a step 216 can include data for extensions 217a below. Device 101 could also use cryptographic algorithms 101s with the selected parameters from a step 216 in order to generate a “client hello” message 218. Device 101 could also store first list of cryptographic algorithms 101s and a second list of cryptographic parameters 101x for device 101 and generate extensions 217a using the first list and the second list.

In a step 101q', device 101 can generate device ephemeral private key eSK.TLS.Device 101q and the corresponding device ephemeral public key ePK.TLS.Device 101n using the selected set of cryptographic parameters from a step 216 above. The generation of ePK.TLS.Device 101n and eSK.TLS.Device 101q were also previously described above for device 101 in Figure 1. Note that the ephemeral private and public key for device 101 and the ephemeral private and public key for network 103 can support post-quantum cryptographic algorithms, as specified by cryptographic algorithms and

cryptographic parameters stored in the two nodes. At the conclusion of a step 101q', device 101 can conduct a step 217 in order to complete the processing or generation of the "client hello" message 218. The "client hello" message 218 generated in a step 217 can include a random number generated by device 101 using cryptographic algorithms 101s, a list of cipher suites supported by device 101 such as a symmetric ciphering algorithm and a cipher block chaining mode, the identity or data for a key derivation function including a hash function, and the list of supported extensions 217a according to section 4.2 of IETF RFC 8446. In exemplary embodiments, the data for a "client hello" message 218 can be generated in a step 217 using the set of cryptographic algorithms 101s in device 101 and parameters from device provider certificate Cert.Provider 104c.

Device 101 can process extensions 217a, where the extensions 217a can either (i) omit a certificate_authorities extension or (ii) include at least a certificate authority specified by cert.provider 104c as stored by device 101, or (iii) the identity for the root certificate issuer 105, where device 101 includes a root certificate cert.root 105a as depicted in Figure 1 above. For some exemplary embodiments, a device 101 can verify a subsequent certificate received from AUSF 103b as cert.provider 104c by simply comparing if the received cert.provider 104c from AUSF 103b matches the cert.provider 104c stored in memory 101m, as depicted in Figure 1 above. The previous sentence pertains to verifying a received cert.provider 104c from AUSF 103b is trusted, and a device 101 would use the public key in cert provider 104c in order to verify the digital signature in a "CertificateVerify" 222a below segment or data within a TLS handshake with MNO 102 and AUSF 103b. In this manner the verification through a certificate authority chain could be omitted in exemplary embodiments, thereby simplifying mutual authentication for device 101 and AUSF 103b/MNO 102. Note that device 101 may not have access to IP network 107 other than sending messages to SEAF 102b, and thus device 101 may not have the ability to verify a server certificate from AUSF 103b (different than device provider certificate 104c) that does not verify through an existing root certificate or certificate authority previously stored in device 101, if the received certificate from AUSF 103b does not have a chain terminating with a stored root certificate 105a.

Note that Cert.Provider 104c and device certificate Cert.Device101c can be stored by device 101 before device 101 sends "client hello" message 218, and also in a manner under the control of device provider 104 (including during any of device configuration, device manufacturing, or device distribution), and thus device provider 104 can take steps when processing device provider certificate Cert.Provider 104c and device certificate Cert.Device101c to ensure the specified cryptographic parameters within the certificates are compatible. Device provider certificate Cert.Provider 104c and device certificate Cert.Device101c can have compatible or equivalent sets of cryptographic parameters (such as specifying the same digital signature algorithm and mutually supported or equivalent named elliptic curves), and in this manner device 101 and AUSF 103b can be reasonably assured that (i) the

negotiation and use of cryptographic parameters in TLS messages such as messages 218 and 227 will be compatible and (ii) the negotiation of a TLS handshake in messages 218 and 227 will be successful.

Device provider can select or specify for device 101 cryptographic parameters including extensions in section 4.2 of TLS version 1.3 for cryptographic algorithms 101s (such as during a device configuration step to load device 101 with the data depicted and described in Figure 1). The selected or specified cryptographic parameters can be compatible with device provider certificate Cert.Provider 104c and device certificate Cert.Device101c. In this manner, a TLS session can be reasonably assured to successfully be negotiated between device 101 and AUSF 103b before the series of steps and message flows depicted in Figure 2, including a message 218, a message 227, and a message 233.

In other words, through the use of (i) device provider certificate Cert.Provider 104c for AUSF 103b as a server certificate and (ii) cryptographic algorithms 101s supporting device provider certificate Cert.Provider 104c for AUSF 103b, a TLS session can be assured to be successful. This contrasts with conventional technology, where if AUSF 103b selects and uses a server certificate not stored in device 101 then the negotiation of the TLS session is more likely to fail from both (i) incompatible sets of cryptographic parameters or cryptographic algorithms, and (ii) the server certificate may not be able to be verified through a certificate chain in device 101 up to a stored root certificate Cert.CI-root 105a, and (iii) device 101 may not have access to IP network 107 in order to obtain additional certificates that could verify a server certificate through a stored root certificate.

If a server certificate selected by AUSF 103b for use in a TLS session is both (i) not previously securely received and stored by device 101, and (ii) is authenticated or verified through a certificate chain with a different root certificate than Cert.CI-root 105a stored by device 101, then device 101 may not be able to verify the sever certificate selected by AUSF 103b (different than Cert.Provider 104c stored by device 101). The potential issues for AUSF 103b to select a server certificate different than device provider certificate Cert.Provider 104c stored by device 101 are compounded by the likely scenarios where (i) hundreds of different AUSF 103b may be used globally with different cryptographic parameters and cryptographic algorithms used for the server certificate and also (ii) server certificates with different verifying certificate chains up to different root certificates. The steps and message flows depicted in Figure 2, along with supporting data recorded as depicted in Figure 1, provide a solution to the significant number of issues that can arise when an AUSF 103b uses a server certificate different than device provider certificate Cert.Provider 104c. In summary to solve the many potential issues and problems of an AUSF 103b using a server certificate different than the device provider certificate Cert.Provider 104c stored by device 101, the device 101 can (i) store the device provider certificate Cert.Provider 104c before sending a SUCI 101b in a message 205, (ii) the AUSF 103b can receive the device provider certificate Cert.Provider 104c from a device provider 104 in a message 213, and (iii) use the device provider certificate Cert.Provider 104c when conducting a TLS session with device 101 as depicted in Figure 2.

Device 101 can then send the TLS “client hello” message 218 to MNO 102 and through SEAF 102b, where MNO 102 forwards the “client hello” message 218 to network 103 and AUSF 103b. The “client hello” message 218 through MNO 102 is also depicted as message 8 and 9 in Figure B.2.1.1-1 of 3GPP TS 33.501 version 15.1.0. A “client hello” message 218 can include a key share value of ePK.TLS.Device 101n, which is depicted in Figure 2. A “client hello” message 218 can include additional data for a “client hello” message as described in section 4.1.2 of IETF RFC 8446, including the data processed by device 101 in 216 and 217, such as selected extensions 217a. Note that MNO 102 and SEAF 102b can include a Nausf_UEAuthentication_Authenticate Request value with the message 218a, such that AUSF 103b can determine which device 101 the “client hello” message 218 is associated with. In exemplary embodiments and also as supported by 3GPP TS 33.501 version 15.1.0, the Nausf_UEAuthentication_Authenticate Request received with message 218a by AUSF 103b can include the SUCI 101b. As depicted in Figure 2, the SEAF 102b can include the SUCI 101b with the “client hello” in message 218a sent to AUSF 103b that includes the “client hello” message 218 from device 101.

AUSF 103b can receive message 218a and conduct a series of steps to process the “client hello”. AUSF 103b can conduct a verification 219a of the received ePK.TLS.Device 101n using the named curve specified by cryptographic parameters within the “client hello” message 218 within extensions 217a. The verification step can comprise verifying that the received public key is on the curve specified by the received cryptographic parameters in extensions 217a, the public key value does not comprise the “point at infinity”, etc. The verification step 219a can comprise the verification step within section 4.2.8.2. of IETF RFC 8446.

AUSF 103b can then conduct a step 219b in order to select at least device provider certificate Cert.Provider 104c from a network database 103d using the SUCI 101b received along with the “client hello” from SEAF 102b in a message 218a. As noted above in a step 214 for AUSF 103b, the AUSF 103b can previously store device provider certificate Cert.Provider 104c, device certificate Cert.Device 101c, SUPI 101a, and SUCI 101b for device 101 in network database 103d after receiving the message 213 above from device provider 104. A step 219b can also comprise AUSF 103b also selecting the device certificate Cert.Device 101c from a network database 103d using the SUCI 101b received along with the “client hello” from SEAF 102b, although the device certificate Cert.Device 101c could be selected at a later time, such as after the receipt of a message 233. An AUSF 103b can use a step 219b to select a device provider certificate 104c from a network database 103d using the SUCI 101b from message 218a for conducting an EAP-TLS authentication. The selected device provider certificate 104c from a step 219b can comprise the server certificate for an AUSF 103b to use when conducting an EAP-TLS authentication with device 101, such as the selected device provider certificate 104c comprising the server certificate below in a “server hello” within message 227.

At step 219c, AUSF 103b can further process the “client hello” by taking the steps specified in IETF RFC 8446, such as generating a server random number, selecting extensions 219d according to section 4.2 of RFC 8446 based on a subset of extensions 217a received in the “client hello” message 218, generating a certificate request 219e according to section 4.3.2 of RFC 8446, etc. The certificate request 219e from a step 219c can comprise a request for device 101 to mutually authenticate with AUSF 103b using at least a device certificate 101c. In exemplary embodiments, for security and commercial reasons, a device provider 104 may prefer to record the secret key for the public key in device provider certificate Cert.Provider 104c. For these embodiments, then AUSF 103b may not be able to internally generate the digital signature for a certificate verify according to section 4.4.3 of RFC 8446. AUSF 103b can then send device provider 104 a message 220, where message 220 include “key share” 103n, which comprises ePK.TLS.Network 103n, Extensions 219d, CertificateRequest 219e, and device provider certificate cert.provider 104c. Note that message 220 can also include identifying information for data within the message 220, such as a SUPI 101a (depicted) or a SUCI 101b, such that device provider 104 could track which device 101 the data within message 220 is associated with.

In exemplary embodiments, a CertificateRequest 219e processed by AUSF 103b can either (i) omit a certificate_authorities extension or (ii) use at least a certificate authority specified by cert.device 101c recorded for device 101 in a network database 103d and also recorded by device 101. An AUSF 103b can verify a subsequent certificate received by device 101 as cert.device 101c by comparing if the received cert.device 101c from device 101c matches the cert.device 101c received from device provider 104 in a message 213, and in this manner the verification through a certificate authority chain could be omitted in exemplary embodiments thereby simplifying mutual authentication for device 101 and AUSF 103b/MNO 102 while maintaining security. The sentence above pertains to verifying a received cert.device 101c from device 101 is trusted, and an AUSF 103b would use the public key in cert.device 101c in order to verify the digital signature in a “CertificateVerify” segment within a TLS handshake with device 101. “CertificateVerify” in Figure 2 and as described herein can comprise a “CertificateVerify” according to section 4.4.3 of RFC 8446.

Note that a message 220 from AUSF 103b to device provider 104 could omit device provider certificate cert.provider 104c since device provider 104 previous sent 104c to AUSF 103b, however including all the depicted elements of a message 220 in one package or string of data segments can help ensure the input for a digital signature by device provider 104 will match exactly the bit sequence and message/segment structure as the “message to sign” input into a digital signature algorithm operated by device provider 104 with SK.Provider 104e in order to generate below a “CertificateVerify” 222a digital signature. In other words, in exemplary embodiments, the contents of a message 220 can include all of the data for AUSF 103b to generate and send a “CertificateVerify” 222a digital signature in a subsequent “server hello” message 227.

Device provider 104 can receive the message 220 with identifying information for device 101 such as the SUPI 101a (depicted) or a SUCI 101b, and in a step 221 verify the selected extensions 219d are supported by device 101 using data recorded by a device 101 in a device database 101d. Device provider 104 could compare that selected extensions 219d are supported by cryptographic algorithms 101s and cryptographic parameters 101x stored in device 101. Note that although not depicted for a device database 101d in Figure 3 below, a device database 104d could store a first list of cryptographic algorithms 101s and a second list of cryptographic parameters 101x in a device database 104d and in a step 221 determine that the selected extensions 219d are supported by device 101 by querying for the first list and the second list in the device database 104d using the SUPI 101a or other identifying information for device 101 in message 220. In a related exemplary embodiment, device provider 104 could send network 103 and AUSF 103b the first list of cryptographic algorithms 101s and a second list of cryptographic parameters 101x for device 101 using SUCI 101b in a message 213 above. Network 103 could use the first list and the second list in order to select extensions 219d in a step 219b, where the selected extensions 219d can be (i) the set of extensions 217d from a “client hello” message 218 or a (ii) a subset of the extensions 217d.

Step 221 can also determine that the CertificateRequest 219b can be properly processed by device 101, such as including a certificate_authorized extension with at least one certificate authority certificate stored in device 101, such as the certificate authority specified for cert.device 101c. At step 222, device provider can generate a digital signature comprising a “CertificateVerify” for the data received in a message 220 using the secret key SK.Provider 104e corresponding to the public key in Cert.Provider 104c and also the cryptographic parameters for the digital signature as specified in Cert.Provider 104c or extensions 219d. The digital signature processed in a step 222 can comprise a “CertificateVerify” 222a. In a step 222, device provider 101 could select the secret key SK.Provider 104e from a device database 104d using the device provider certificate cert.provider 104c received in a message 220. As contemplated herein, the certificate 104c received in a message 220 could also comprise a unique certificate identity for the certificate 104c, such that device provider 104 can identify the device provider certificate cert.provider 104c using the unique certificate identity.

Device provider 104 can then send AUSF 103b a message 223 with a digital signature of “CertificateVerify” 222a, where the “CertificateVerify” 222a can be over the data sent by AUSF 103b in a message 220 (e.g., “key share” 103n, which comprises ePK.TLS.Network 103n, Extensions 219d, CertificateRequest 219c, and device provider certificate cert.provider 104c). In this manner from the combinations of a step 219c, message 220, and step 222, a AUSF 103b can (i) generate a “message to sign” for a “server hello” comprising key_share 103n, Extensions 219d, CertificateRequest 219b, Certificate 104c, and (ii) receive a digital signature of “CertificateVerify” 222a from device provider 104 over the “message to sign”. AUSF 103b can use the received digital signature of a “CertificateVerify” 222a in order to generate a complete “server hello” message 227 for a SUCI 101b

that could be successfully processed by device 101 using the authentication and encryption steps specified for TLS 1.3 in IETF RFC 8446.

Although Figure 2 depicts device provider 104 generating the digital signature for “CertificateVerify” 222a, in some exemplary embodiments (i) an AUSF 103b could generate the “CertificateVerify” 222a, and (ii) the messages 220, step 222, and message 223 could be omitted from a system 200. For this embodiment, then device provider 104 could send the private key SK.Provider 104e for device provider certificate Cert.Provider 104c through secure session 202, such as in a message 213 above. For this embodiment, a network 103 could store the private key SK.Provider 104e for device provider certificate Cert.Provider 104c in a network database 103d and select the private key SK.Provider 104e using the selected device provider certificate Cert.Provider 104c for a device 101 SUCI 101b, where the selection of a device provider certificate Cert.Provider 104c for a device 101 SUCI 101b was described above for a step 219b.

AUSF 103b can receive message 223 and conduct steps to generate a “server hello” response for the received “client hello” from device 101. In step 224, AUSF 103b could verify the digital signature for “CertificateVerify” 222a using the public key and parameters in device provider certificate 104c in order to confirm the digital signature is correct and can be properly processed by device 101 with the public key and parameters in device provider certificate 104c.

For some exemplary embodiments of a system 200, a network 103 could use a step 219b to select a server public key for use in an EAP-TLS authentication with device 101 in a subsequent “server hello” message 227 instead of using a received device provider certificate 104c. Network 103 could record and use a corresponding server private key for the server public key. For these embodiments, a network 103 or AUSF 103b could send a server certificate for the selected server public key in a message 220, instead of the data depicted for a message 220 in Figure 2. For these embodiments, a message 220 could comprise a “certificate signing request” for the server public key in the selected server certificate. The “certificate signing request” from network 103 for the selected server certificate/server public key could comprise a request as described in the Wikipedia article titled “Certificate Signing Request” from May 1, 2019, which is herein incorporated by reference in its entirety. Device provider 104 could receive the “certificate signing request” for the server public key in the server certificate in the message 227. Device provider could generate a new certificate for the server public key by signing with the device provider private secret key SK.Provider 104e. The device provider secret key SK.Provider 104e could be selected from a device database 104d using the realm portion of a SUPI 101a in a message 220. For these embodiments, device provider 104 could send network 103 and AUSF 103b the new certificate in a message 223. Network 103 could then include the new server certificate in a subsequent “server hello” message 227, where the “CertificateVerify” 222a could be a digital signature generated by network 103 using the server private key for the server public key in the new certificate, where the new certificate could be received from device provider 104 in response to the “Certificate Signing

Request”. For these embodiments, a device 101 could (i) verify a “server hello” message using the server public key in the new server certificate, and (ii) verify the new server certificate using the device provider certificate 104c stored in device 101, as depicted in Figure 1.

At step 225, AUSF 103b can conduct an ECDH key exchange with device ephemeral public key ePK.TLS.Device 101n received in message 218 and server ephemeral private key eSK.TLS.Network 103q derived in a step 103q’ in order to mutually derive shared secret 225a and then derive a symmetric ciphering key 225b and an associated MAC key. A step 225 can follow the steps and procedures as specified by section 7 of RFC 8446 for the shared secret 225a and then deriving the symmetric ciphering key 225b, although other possibilities exist as well for the derivation of a shared secret key 225a and a symmetric ciphering key 225b without departing from the scope of the present invention. As supported by section 7 of RFC 8446, (i) a first portion of a symmetric ciphering key 225b and an associated MAC key for encryption by AUSF 103b and decryption by device 101 can be different than (ii) a second portion of a symmetric ciphering key 225a and an associated MAC key for decryption by AUSF 103b and encryption by device 101. However, both device 101 and AUSF 103b can conduct a step 225 in order to derive both the first and second portions of a symmetric ciphering key 225b.

At step 225, AUSF 103b can derive a shared secret 225a with device 101 as a point with x and y coordinates, where the device ephemeral public key is raised to the power of the scalar ephemeral private key with a modulus according to the named curve to for the PKI keys in cryptographic parameters associated with the keys and also in extensions 219d selected in a step 219b. The x coordinate of the resulting point can comprise a shared secret 225a. AUSF 103b or network 103 could store the derived shared secret 225a and the symmetric ciphering key 225b from a step 225 in a network database 103d as depicted in Figure 3 below, along with the SUCI 101b received in a message 218 and/or the combination of SUCI 101b and SUPI 101a received in a message 213. The symmetric key 225b and the associated MAC key can be derived from the shared secret 225a using a key derivation function (KDF) or a secure hash algorithm mutually shared with device 101. The KDF and secure hash algorithm could be specified and shared between device 101 and AUSF 103b in extensions 217b and extensions 219d, such as a secure hash algorithm of SHA-256, SHA-384, etc.

Device 101 can take equivalent steps with the corresponding keys as used by AUSF 103b in order to derive the shared secret 225a and the symmetric key 225b and an associated MAC key, which is also described in section 7 of IETF RFC 8446. At step 226, AUSF 103b can complete the processing of a “server hello” message according to RFC 8446 by encrypting the values Extensions 219d, CertificateRequest 219e, Certificate 104c, OSCP stapling 212a (from message 213), and Certificate Verify 222a with the symmetric ciphering key 225b derived in a step 225.

Other data could be included in a “server hello” message as well without departing from the scope of the present invention. Although Annex B of 3GPP TS 33.501 version 15.1.0 describes the use of TLS version 1.2, a MNO 102 and SEAF 102b and AUSF 103b in a system 200 could support the use

of TLS version 1.3. Or, the sequence of steps and messages in Figure 2 for a system 200 could be adjusted to support TLS version 1.2 without departing from the scope of the present disclosure. Further, future versions of EAP-TLS authentication for 5G networks and 3GPP networks could support TLS version 1.3, where the future versions could be compatible with the steps and message flows depicted for a system 200 in Figure 2.

AUSF 103b can then send through MNO 102 the “server hello” message 227. Note that AUSF 103b can insert or add the SUCI 101b for device 101 external to the “server hello” message 227, such that SEAF 102b and MNO 102 can send the “server hello” message 227 to the correct device 101. In other words, MNO 102 can use the SUCI 101b as external to the “server hello” message 227 in order to route the message 227 to device 101 through base stations and at least one gnodeb with a radio 102r. The data within a “server hello” message is depicted in Figure 2 and also described above with a step 226. The bracket values of “{“ and “}” in for message 227 and other messages or data depicted in Figures herein depict that the ciphertext data is encrypted with the derived symmetric ciphering key, such as key 225b from a step 225 and also verified with a MAC code using a MAC key from a step 225. The depiction of the key 225b next to the brackets for message 227 indicate that the message is encrypted with at least a mutually derived symmetric ciphering key 225b and does not suggest or indicate that the key 225b is transmitted with the message.

Device 101 can receive the “server hello” message 227 and take steps in order to process the message. At step 219a, device 101 can verify the received ephemeral public key using the same steps as a step 219a for AUSF 103b above, which can also comprise the verification step 4.2.8.2 in RFC 8446. At step 228, device 101 can conduct an ECDH key exchange with device ephemeral private key eSK.TLS.Device 101q and server ephemeral public key ePK.TLS.Network 103n derived in a step 103q in order to mutually derive a shared secret and then a symmetric ciphering key and a MAC key. A step 228 can follow the steps and procedures as specified by section 7.4 of RFC 8446. At step 228, device 101 can derive a shared secret with AUSF 103b as a point with x and y coordinates, where the network/server ephemeral public key 103n is raised to the power of the scalar device ephemeral private key with a modulus according to the named curve to for the PKI keys. The x coordinate of the resulting point can comprise a shared secret 225a, which was mutually derived by the AUSF 103b in a step 225. The symmetric key 225b and the MAC key can be derived from the shared secret 225a using a secure hash function, and additional data shared between device 101 and AUSF 103b in messages 218 and 217 could be input into the secure hash function with the shared secret 225a. A step 228 can also comprise device 101 decrypting the ciphertext data from a message 227 using the derived symmetric ciphering key 225b and verifying message 227 integrity with the MAC code and the MAC key derived in step 228.

A step 229 can comprise device 101 verifying the digital signature in the digital signature for “CertificateVerify” 222a using the public key and parameters in device provider certificate 104c in order

to confirm the digital signature is correct. Device 101 could confirm the received value for “CertificateVerify” 222a matches a calculated value by device 101 for “CertificateVerify” 222a. The message or data input into the digital signature algorithm can comprise the data key_share 103n, Extensions 219d, CertificateRequest 219e, Certificate 104c, and OSCP stapling 212a, and also follow the steps for signature verification in RFC 8446. Other possibilities exist as well for the steps to verify a digital signature of a “CertificateVerify” 222a using the public key in a device provider certificate 104c without departing from the scope of the present disclosure.

A step 230 can comprise device 101 verifying that the received device provider certificate 104c matches the recorded device provider certificate 104c in memory 101m as depicted in Figure 1. Or, if device provider certificate 104c does not match the recorded device provider certificate 104c in memory 101m, then device 101 can conduct a series of steps to verify the received device provider certificate 104c up through at least one recorded and securely received certificate within device 101, such as a root certificate Cert.CI-root 105a. Note that in some embodiments, the sequence of steps 229 and 230 could be switched, and the order of other steps depicted as contemplated herein could be adjusted without departing from the scope of the present invention. In addition, a step 230 can comprise device 101 verifying an OSCP stapling 212a for embodiments where device provider certificate 104c does not match the recorded device provider certificate 104c. Further, a step 230 can comprise device 101 verifying an OSCP stapling 212a if the device provider certificate 104c matches the certificate 104c stored by device 101 in order to confirm the current validity status of the device provider certificate 104c (e.g. device 101 could determine that certificate 104c has not been revoked).

Note that the system 200 depicted in Figure 2 can avoid the requirement that device 101 needs to verify OSCP stapling 212a, for embodiments where device provider 104 generates “CertificateVerify” 222a in a step 222. The presence of a “CertificateVerify” 222a, where device provider 104 generates the “CertificateVerify” 222a in a step 222 can confirm the certificate cert.provider 104c is valid for device 101. If the cert.provider 104c was revoked, then device provider 104 and/or AUSF 103b could decline to provide “CertificateVerify” 222a to device 101. If another party than device provider 104 obtained the secret key SK.Provider 104e, the party would not feasibly be able to use the key since MNO 102 only sends SUCI 101b for device 101 to AUSF 103b, and thus device 101 would not authenticate with a different AUSF 103b as specified in the realm of the SUCI 101b.

At step 231, device 101 can select a device certificate 101c that is compatible with the parameters received within message 228, such as with extensions 219d. A device 101 could record a plurality of different device certificates 101c, such as supporting different cryptographic parameters for digital signature algorithms and also different certificate authorities. Device 101 could use the data for extensions 219d, such as a certificate authority specified in extensions 219d, in order to select device certificate 101c. In some exemplary embodiments, the certificate authority for device certificate 101c

can be device provider 104, where the digital signature for device certificate 101c is signed with the device provider certificate issuer secret key SK.Provider-CI 104g. In other words, in exemplary embodiments device 101 selects a device certificate 101c in a step 231 where the signature authority for device certificate 101c is the device provider certificate issuer for device provider 104. Or, in other embodiments, device 101 could record a single certificate 101c and AUSF 101c can verify the certificate 101c by comparing the certificate 101c received from device 101 with the certificate 101c received from device provider 104.

At step 232, device 101 can generate a digital signature for a "CertificateVerify" 232a over at least the selected device certificate 101c with the private key SK.Device 101d for the selected device certificate 101c using the parameters from extensions 219d in message 227. The private key SK.Device 101d is depicted and described above in connection with Figure 1. Device 101 could follow the steps for generating a "CertificateVerify" in section 4.4.3 of RFC 8446, and other possibilities exist as well without departing from the scope of the present disclosure. A step 232 can include device 101 encrypting the device certificate 101c and the "CertificateVerify" 232a into a ciphertext with the symmetric ciphering key 225b derived in a step 228 above, as well as generating a MAC code using the derived MAC key.

Device 101 can then MNO 102 a message 233, where message 233 includes the ciphertext from a step 232 with the encrypted device certificate 101c and the encrypted digital signature of "CertificateVerify" 232a. Note that MNO 102 can use the radio resource connection of device 101 in order to track or identify device 101, and MNO 102 using SEAF 102b can append the SUCI 101b originally sent by device 101 in a message 205 to the message 233, equivalent to MNO 102 or SEAF 102b adding SUCI 101b to message 218 above. In other words, MNO 102 can track the radio resource connection used by device 101 in order to (i) append or add SUCI 101b to message 218 (e.g. message 218a received by AUSF 103b) (ii) append or add SUCI 101b to message 233 received by AUSF 103b. In other words, AUSF 103b can receive the message 233 with the SUCI 101b added, although device 101 may not add SUCI 101b to message 233. Or, in some exemplary embodiments device 101 can add SUCI 101b to both message 218 and message 233. As described below, AUSF 103b can use a network database 103d to record data for sessions with device 101, such as the SUCI 101b and the device provider certificate 104c as well as the server ephemeral private key eSK.TLS.Network 103q and symmetric ciphering key 225b associated with a device 101 and TLS messages. As depicted for a network database 103d in Figure 3 below, a network 103 could also store the derived symmetric ciphering key 225b from a step 225, and MAC key as well (although a MAC key in database 103d is not depicted in Figure 3), in order to select symmetric ciphering key 225b using the SUCI 101b from MNO 102 and decrypt the ciphertext in a message 233.

In exemplary embodiments, the device provider certificate 104c received from device provider 104 by AUSF 103c can function as the server certificate for AUSF 103b with device 101. AUSF 103b

can use a plurality of different server certificates with a plurality of different devices, and the proper certificate for AUSF 103b to use could be tracked in a network database 103d and selected for a device 101 from messages via MNO 102 using the SUCI 101b added to the messages by MNO 102.

In a step 234, AUSF 103b can use the selected (x) symmetric key 225b from a network database 103d for (y) SUCI 101b received in a message 233 from MNO 102 in order to decrypt the ciphertext included in message 233. The plaintext values from a message 233 can comprise the device certificate 101c and the "CertificateVerify" 232a. "CertificateVerify" 232a can comprise a digital signature from device 101 over the plaintext data within message 233 (but potentially excluding the digital signature for "CertificateVerify" 232a. At step 234, AUSF 103b can compare that received device certificate 101c from device 101 is the same as the received device certificate from device provider 104 in a message 213 above. AUSF 103b can trust or verify the received device certificate 101c from device 101 if the received device certificate 101c from device 101 in a message 233 is the same as the device certificate 101c received from device provider 104, such as in a message 213 via secure session 202. Or, if the received device certificate 101c from device 101 is not the same as 101c from device provider 104 (or AUSF 103b does not receive a device certificate 101c from device provider 104), then AUSF 103b can verify cert.device 101c through a certificate chain up to a securely recorded certificate in AUSF 103b such as a root certificate Cert.CI-root 105a.

At step 235, AUSF 103b can verify the digital signature value for "CertificateVerify" 232a using the device public key in cert.device 101c as well as the signature parameters in cert.device 101c. AUSF 103b can confirm the received value for "CertificateVerify" 232a matches an internally calculated value for "CertificateVerify" 232a. After successful device authentication in a step 235, AUSF 103b can conduct a step 236 in order to generate an anchor key for SEAF 102b and MNO 102. With EAP-TLS authentication between device 101 and AUSF 103b, the AUSF 103b and device 101 can mutually derive a key Kausf using the steps described in Annex B, section B.3 titled "Key Derivation" for 3GPP TS 33.501 version 15.1.0. AUSF 103b can conduct the steps to derive Kausf in a step 236. Although section B.3 describes the derivation of a Kausf in the context of EAP-TLS authentication using TLS version 1.2 with IETF RFC 5216, the equivalent steps can be conducted with a future version of EAP-TLS authentication supporting TLS version 1.3.

In summary, the successful handshake of messages 218, 227, and/or 233 can be used in a step 236 to derive a TLS master secret as described in section 7.1 of RFC 8446. Data input into the derivation of a TLS master secret for AUSF 103b can comprise the shared secret 225a derived in a step 225 and the plaintext values for data transmitted in message 218 and 227. The shared secret 225a derived in a step 225 and the plaintext values for data transmitted in message 218 and 227 can be input in a step 236 into a key derivation function using a secure hash algorithm in order to derive a TLS master secret for AUSF 103b.

According to the EAP-TLS authentication for RFC 5246, the derived TLS master secret can be input into a key derivation function such as described in section 2.3 titled “Key Hierarchy” of RFC 5246 in order to generate both a master session key (MSK) and an extended master session key (EMSK) 236b. For some exemplary embodiments, the extended master session key 236b can comprise a length of 512 bits. For exemplary embodiments, including those supporting 3GPP TS 33.501 versions 15.1.0, the first 256 bits of the EMSK 236b can be taken as the value for K_{ausf} in a step 236 in order to select an anchor key 236a which could comprise K_{ausf}. Note that (i) future versions of EAP-TLS authentication that explicitly support TLS version 1.3 can also derive a value of EMSK 236b from the TLS master secret as described in section 7.1 of RFC 8446, and (ii) future versions of 3GPP TS 33.501 can specify the derivation of K_{ausf} from the EMSK 236b.

In exemplary embodiments, a step 236 can comprise (i) AUSF 103b mutually deriving a TLS master secret according to RFC 8446 with device 101, and then (ii) AUSF 103b mutually deriving with device 101 an anchor key such as K_{ausf} using the mutually derived TLS master secret. An intermediate step between (i) and (ii) for both AUSF 103b and device 101 can be deriving an EMSK 236b according to IETF EAP-TLS standards, where the K_{ausf} is derived from the EMSK 236b. Other possibilities for AUSF 103b to mutually derive a shared secret key (different than key 225a, but using at least key 225a) in a step 236 using data from a TLS session such as the keys and values within messages 218, 227, and/or 223 are possible as well without departing from the scope of the present invention.

After a step 236, AUSF 103b can send MNO 102 and SEAF 102b a message 237 through secure session 201, where message 237 can include both the mutually derived (between AUSF 103b and device 101) anchor key 236a, which could comprise a K_{ausf}, and the value of SUPI 101a. Note that anchor key 236a can comprise a derived shared secret key that is mutually derived between AUSF 103b and device 101, and a different name than an “anchor key” of “K_{asuf}” could be used to describe the key 236a, such as a shared secret, a derived shared secret, a shared secret key, etc. Or, in some exemplary embodiments an AUSF 103b could send the TLS master secret or EMSK 236b in a message 237, and an SEAF 102b could derive anchor key 236a from the TLS master secret or EMSK 236b received in a message 237. The SEAF 102b can receive message 237 and conduct a step 238 in order to process the message. In a step 238, SEAF 102b can conduct additional key derivations using the anchor key 236a such as K_{asuf} in order to derive an associated key set 238a K_{seaf}, K_{amf}, K_{gnb}, etc., as specified in 3GPP TS 33.501.

As depicted in Figure 2, device 101 in a step 239 can conduct the equivalent steps as AUSF 103b in a step 236 in order to mutually derive a shared secret key such as an anchor key 236a or K_{ausf} using at least the TLS master secret. Device 101 can use the steps specified in section 7.1 of TLS version 1.3 RFC 8446 in order to derive the TLS master secret. Data input into the derivation of a TLS master secret for device 101 can comprise the shared secret 225a derived in a step 228 and the plaintext values for data transmitted in message 218 and 227 (e.g. after decryption of a message 227). The shared

secret 225a derived in a step 228 and the plaintext values for data transmitted in message 218 and 227 can be input into a key derivation function in order to derive a TLS master secret for device 101. The key derivation function can comprise a secure hash algorithm. Device 101 could take additional steps equivalent to those described in a step 236 above for an AUSF 103b to derive an EMSK 236b using at least derived TLS master secret. Corresponding to a step 236 for AUSF 103b above, a device 101 could derive an anchor key 236a from the EMSK 236b using standards specified for EAP-TLS authentication and/or 3GPP standards for 5G networks. A step 239 for device 101 can also comprise device 101 deriving an associated key set 239a using a derived Kausf from the TLS master secret for communication with MNO 102, such as a Kseaf, Kamf, Kgnb, etc. as specified in 3GPP TS 33.501.

After (i) a step 239 by device 101 to derive a shared secret key such as Kausf and related keys such as those specified in 3GPP TS 33.501 and (ii) a step 238 by MNO 102 with SEAF 102b to also use the shared secret such as Kausf received in a message 227 to derive related keys such as those specified in 3GPP TS 33.501, then device 101 and MNO 102 can exchange encrypted and integrity protected messages in RF communications 240. RF communications 240 can be conducted with keys 239a derived by device 101 in a step 239 and keys 238a derived from an anchor key by SEAF 102b in a step 238.

The steps for encrypting messages and also conducting integrity checks of messages in communications 240 can also be specified in 3GPP TS 33.501 and related standards. In addition, for embodiments where MNO 102 and device 101 use an 802.11 network instead of a 5G network, then device 101 and MNO 102 could use at least a portion of an EMSK 236b (derived from TLS master secret) in order to derive related encryption and integrity protection keys such as, but not limited to a pairwise transit key (PTK) in order to support the 802.11 network between device 101 and MNO 102. In summary, the encryption and integrity protection of messages in communications 240 are with keys derived from the TLS master secret, where the derivation of the TLS master secret is depicted and described above in a step 236 for AUSF 103b and a step 239 for device 101. In addition, the encryption and integrity protection of messages in communications 240 are with keys derived from shared secret 225a in a step 225 by AUSF 103b and the same shared secret 225a in a step 228 by device 101, since the TLS master secret for both nodes uses the shared secret 225a and data available to both device 101 and AUSF 103b such as the plaintext data in messages 218 and 227.

Note that the message 218 can include a random number generated by device 101 and the message 227 can include a random number generated by AUSF 103b, where the random numbers are also used in the derivation of the TLS master secret. In this manner, and by (i) including the random number generated by device 101 in a message 218 and the random number generated by AUSF 103b in a message 227, and (ii) by including the data from a message 218 and a message 227 in the derivation of a TLS master secret, additional entropy or random values can be used by AUSF 103b and device 101 in addition to mutually derived shared secret 225a for the derivation of a TLS master secret. As

described above, the TLS master secret can be used to derive a shared secret or an anchor key 236a for device 101 and AUSF 103b and MNO 102 to use in subsequent communications such as a connection for messages 240. In other words, at least a first random number within a “client hello” message 218 and a second random number within a “server hello” message 227, along with the shared secret 225a can be used by device 101 and AUSF 103b to derive a shared secret for subsequent communications 240, where the shared secret can comprise an anchor key 236a. For 802.11 networks between device 101 and MNO 102, the shared secret can comprise at least a portion of EMSK 236b.

For some exemplary embodiments, the wireless networking technology supported by MNO 102 can comprise WiFi networking supporting IEEE 802.11 standards. For these embodiments, the anchor key 236a sent in a message 237 through secure session 201 to MNO 102 can comprise any of (i) the shared secret key 225a, (ii) the TLS master secret derived in a step 236, and/or (iii) the value of EMSK 236b. In other words, when supporting WiFi networking for EAP-TLS authentication for device 101 and MNO 102, network 103 could send a mutually derived anchor key 236a that is different than a Kasuf as contemplated by 5G standards in 3GPP TS 33.501. Other possibilities exist as well for the structure or contents of a mutually derived anchor key 236a sent in a message 237 from network 103 to wireless network 102 without departing from the scope of the present invention. Further, when supporting 802.11 wireless networks for device 101 and MNO 101, the use of a SUCI according to 3GPP TS 33.501 can be optional, and the device identity as received by network 103 in a message 205 can comprise a SUPI 101a without encryption and also without the device ephemeral public key ePK.ECIES.Device 101h.

Figure 3

Figure 3 is an illustration of a device database and a network database with exemplary data stored, in accordance with exemplary embodiments. A device provider 104 could store and operate a device database 104d as depicted and described in connection with Figure 1. Network 103 could store and operate a network database 103d in connection with Figure 1. The device database 104d or network database 103d could be recorded in a server 111 or collection of servers securely connected to device provider 104 or network 103, respectively. Each database could operate using software such as Oracle, Microsoft SQL Server, MySQL, and other possibilities exist for a database without departing from the scope of the present disclosure.

Although both a device database 104d and network database 103d depict multiple values, strings, or numbers stored in each column and row, some values could be omitted at certain points in time, such as the data not being available at a point in time, although the omitted data could be provided or available later. For example, with a network database 103d, upon receipt of a message 205 with a new SUCI 101b not previously stored in a network database 103d, network 103 could insert a new row into a network database 103d with SUCI 101b entered and the other data empty, until later received in

later messages (i) such as message 213 from device provider 104 or (ii) received from device 101 via MNO 102. Both device database 104d and network database 103d can be encrypted with a symmetric ciphering key such that the data is stored in physical memory such as storage 111b as ciphertext.

Values depicted in Figure 3 with different numeral designations such as “-1”, “-2”, “-3”, etc. can represent different numbers or strings for the depicted value such as SUPI 101a. Values depicted in Figure 3 with the same numeral designation such as the first three rows of Provider ID 104a in a device database 104d being “104a-1”, then “104a-1”, then “104a-1” can represent that the number or string for provider ID 104a is the same for each of the devices in the first three rows. Values depicted in Figure 3 with different numeral designation such as the last three rows of Provider ID 104a in a device database 104d being “104a-2”, then “104a-2”, then “104a-2” can represent that the number or string for provider ID 104a is different than each of the devices in the first three rows, which are depicted with a designation being “104a-1”.

Device database 104d can record data for a plurality of devices 101 which could be managed or supported by device provider 104. A device database 104 could store values, numbers, or strings for a device number or index, a user name for a network access identifier (NAI), a device identity as a SUPI 101a, a device certificate 101c, a provider identity 104a, a provider certificate 104c, a provider secret key 104e, ECIES secret key 104b, and ECIES parameters 101f. Data within a device database 104d could be recorded or stored before device 101 establishes an initial radio resource connection with MNO 102, such as during device manufacturing or distribution or a device configuration, and other possibilities exist as well for the time before a message 205 when data for device database 104d is recorded. As additional devices are added to device provider 104, then additional rows for the additional devices could be inserted into a device database 104d.

As depicted in Figure 3, a device database could store additional data for a device 101 than the fields or values depicted. Additional data could include the ECIES ephemeral public key 101h received in a message 209 by device provider 104, the SUCI 101b used by a device 101 or a device user name associated with a device 101. In addition, although not depicted for a device database 104d in Figure 3, a device database 103d could also store a first list of cryptographic algorithms 101s and a second list of cryptographic parameters 101x for a device 101 with SUPI 101a in a device database 104d, where the first list and the second list could be used to determine extensions 219d from an AUSF 103b are supported by device 101. The first list and the second list could also be sent to network 103 in message 213 in Figure 2 above. Other possibilities exist as well for data stored or recorded in a device database 104d without departing from the scope of the present disclosure.

Although a single device database 104d is depicted in Figure 3, a device database 104d could comprise either multiple databases or multiple tables with data equivalent or similar to that depicted in Figure 3. In one exemplary embodiment, a device provider 104 could operate multiple device databases 104d, where a first device database 104d is associated with a first network 103 and a second device

database 104d is associated with a second network 103, etc. The realm values for a NAI within a SUCI or SUPI could include a first value for the network portion and/or provider portion of the realm for the first network and first device database. The realm values for a NAI within a SUCI or SUPI could include a second value for the network portion and/or provider portion of the realm for the second network and second device database.

The user name in a device database can comprise the user name portion of a network access identifier (NAI) for a SUPI 101a. The SUPI 101a stored in a device database 104d can comprise the SUPI 101a as described above for a device 101 in Figure 1. Note that the user name can be plaintext with a SUPI. In addition, a “user name” can comprise an identity for a device 101 and a “user name” does not need to be associated with a person. Device certificate 101c stored in a device database 101d can comprise the device certificate 101c stored and used by device 101 as described for a device 101 in Figure 1. The format and data structure of device certificate 101c in a network database 103d, device provider database 104d, and device 101 could take many forms such as a *.pem structure, raw text, *.crt, or *.der. *.crt can comprise a certificate format and *.der can comprise distinguished encoding rules for the certificate. Other certificates described herein, including device provider certificate 104c could be stored or transmitted in any of the exemplary formats described in the previous sentence.

The provider ID 104a stored in a device database 104d can comprise the provider identity used in the realm portion of a SUCI 101b and SUPI 101a for a device 101. The provider ID 104a in a device database 104d can also comprise the identity for a device provider stored in a provider certificate 104c, such as an identity in the “common name” field or also possibly a portion of the “subject alternative name”, and other possibilities exist as well for the location of the provider ID 104a within provider certificate 104c. Note that the provider ID 104a in a SUCI 101b or SUPI 101a is not required to equal or be the same as a provider ID 104a in the provider certificate 104c. A database 104d can store the provider ID 104a in the provider certificate 104c and also the different provider ID 104a in the SUPI 101a. Provider certificate 104c can be stored in device database 104d, network database 103d, and device 101 in a format or structure similar to device certificate 101c described in the paragraph above. In some exemplary embodiments provider certificate 104c can comprise a certificate in *.pem format.

The device provider certificate 104c in a device database 104d and network database 103d can comprise the device provider certificate 104c depicted and described in Figure 1 and Figure 2. As depicted in Figure 3, different devices 101 can use different provider certificates 104c, which is also depicted in device database 104d and network database 103d. Note that the same provider ID 104a can be used with different provider certificates 104c, such as provider ID “104a-2” being associated with both provider certificate “104c-2” and provider certificate “104c-3”. For this embodiment, the same provider ID 104a could be used with two different certificates with different parameters such as different certificate expiration dates or different key lengths or key names or different elliptic curves used for the public keys, etc., such that different devices 101 could use different provider certificates 104c. In

addition, a device 101 using a SUPI 101a could store or be associated with a plurality of different device provider certificates 104c, and the multiple different device provider certificates 104c could be stored for the device 101 in a device database 104d. A specific device provider certificate 104c for the device 101 could be selected based on the network 103 or MNO 102 communicating with device 101, such as parameters supported by the network 103 or MNO 102, where the parameters could be received by device provider 104 via secure session 202.

In exemplary embodiments, the server certificate for a network 103 when conducting an EAP-TLS authentication with device 101 can use the device provider certificate 104c, where the different server certificates used by network 103 and AUSF 103b are selected in a step 219b and identified based on the provider ID 104a in the realm of the SUPI 101a or SUCI 101b. Also note that a device database 104d and a network database 103d could record unique certificate identities for the depicted certificates, and the full certificates as exemplary *.pem files could be stored outside of a database (but identified by the unique certificate identities).

The provider secret key 104e for a provider certificate 104c can be used with the provider public key stored with the device provider certificate 104c. The use of a provider secret key 104e was described above in Figure 1 and Figure 2, such as selecting in a step 222 the provider secret key 104e from a device database 104d based on data in a message 220 such as a device provider certificate 104c for the “message to sign”. Device provider 104 could use the selected device provider secret key 104e in order to conduct the step 222 to create or generate the “Certificate Verify” 222a portion or segment of the TLS “server hello” message subsequent sent by AUSF 103b in a message 227 in Figure 2.

Device database 104d can also store values of ECIES secret key 104b, which corresponds to the ECIES public key 101e stored by device 101. Note that in some exemplary embodiments a device provider 104 can select the ECIES secret key 104b based on the provider ID 104a received within a realm value for the SUCI 101b from a message 209 as depicted in Figure 2. In other words, in exemplary embodiments, there can be a one-to-one mapping or a unique mapping between ECIES secret key 104b and the provider ID 104a in a realm for a device provider 104 in a SUCI 101b from message 209 using a device database 104d. In this manner, a device provider can know which ECIES secret key 104b to use for a plurality of devices 101, where the ECIES secret key 104b to use with an ECIES decryption function 104j can be identified based on the provider ID 104a received in a realm of the SUCI 101b.

Although exemplary data is depicted within a single table within device database 104d in Figure 3, a device database 104d could include multiple different tables. As one example, a separate table could include a “one-to-one” mapping between a provider ID 104a, which could be included in the realm portion of a SUCI 101b, and the value ECIES secret key 104b and the value of ECIES parameters 101f. Note that a device database 104d could also store the of ECIES parameters 101f in a device database 104d, which can specify parameters for use with the ECIES secret key 104b, such as use of

curve 25519 for a first ECIES secret key 104b and the use of curve P-256 for a second ECIES secret key 104b, and other possibilities exist as well.

As depicted in Figure 1, a network 103 could store and operate a network database 103d. Either device database 104d or network database 103d could comprise a collection of servers and storage memory for storing values associated with a plurality of devices in order to support the operation of a system 200 or a system 100. A network database 103d could store values, strings, or numbers for Device identity / SUCI 101b, Device identity / SUPI 101a, Device Certificate 101c, Provider Certificate 104c, TLS Server Ephemeral secret key 103q, TLS Device Ephemeral public key 101n, Shared Secret 225a, Symmetric key 225b, EMSK 236b, and Anchor Key 236a. Additional supporting data for a system 200 could be stored or recorded in a network database 103d as well without departing from the scope of the present disclosure.

Data within a network database 103d could be recorded or stored as network 103 receives data for a device 101 from a MNO 102 and a device provider 104. For example, as a network 103 receives a SUCI 101b for a device 101 from a MNO 102 after device 101 established a radio resource connection with MNO 102 and MNO 102 forwards the SUCI 101b in a message 205 in Figure 2, then the value for SUCI 101b could be added to the network database 103d. As additional data from additional devices 101 is received by network 103, then additional rows for the additional devices could be inserted into a network database 103d. A network database could store additional data or columns for a device 101 than the fields or values depicted.

Additional data in a network database 103d could include the ECIES ephemeral public key 101e received in a message 205 by network 103, and an identity of MNO 102 of MNO ID 102a that sent a message 205. For embodiments where network 103 uses a new server certificate instead of a device provider certificate 104c in the “server hello message” (as described above for an alternative embodiment of message 220 with a “certificate signing request”), then the new server certificate or an identity of the new server certificate could be stored for a device 101 along with the SUCI 101b. Additional data in a network database 103d could include the selected extensions 219d used for generating a “server hello” in message 227, where the extensions 219d could be selected in a step 219c in Figure 2 above and other possibilities exist as well without departing from the scope of the present disclosure.

Although a single network database 103d is depicted in Figure 3, a network database 103d could comprise either multiple databases or multiple tables with data equivalent or similar to that depicted in Figure 3. In one exemplary embodiment, a network 103 could operate multiple network databases 103d, where a first network database 103d is associated with a first device provider 104 and a second network database 104d is associated with a second device provider 104, etc. The realm values for a NAI within a SUCI or SUPI could include a first value for the device provider portion the realm for the first device provider and first device database. The realm values for a NAI within a SUCI or SUPI could include a

second value for the device provider portion of the realm for the second device provider and second device database.

Values for Device identity / SUCI 101b in a network database 103d could be received in a message 205 from MNO 102 in Figure 2. Values in a network database 103d of Device identity / SUPI 101a, Device Certificate 101c, and Provider Certificate 104c could be received in a message 213 in Figure 2. Values for TLS Server Ephemeral secret key 103q could be derived in a step 103q' as depicted and described in connection with a step 103q' in Figure 2. Values for TLS Device Ephemeral public key 101n in a network database 103d could be received in a message 218 from Figure 2. Values for shared secret key 225a and symmetric ciphering key 225b could be derived in a step 225 in Figure 2. Values for EMSK 236b and an anchor key 236a or Kausf could be derived in a step 236 in Figure 2.

Network 103 can use the data recorded in a network database 103d in order to conduct the series of steps and process the series of messages as depicted and described in connection with Figure 2. As one example and as described for a step 219b, network 103 and/or AUSF 103b could use data recorded in a network database 103d with the SUCI 101b value from a message 218a to select at least a device provider certificate 104c as the server certificate to use with EAP-TLS authentication and the generation of a "server hello" in message 227. The values for TLS Server Ephemeral secret key 103q and TLS Device Ephemeral public key 101n, along with associated parameters could be selected in a step 225 using the SUCI 101b from a message 218a in order to conduct a step 225 to generate the shared secret 225a and the symmetric ciphering key 225b. The ciphertext in a "server hello" within message 227 can be encrypted with the symmetric ciphering key 225b and the ciphertext in a message 233 can be decrypted with the symmetric ciphering key 225b from a network database 103d. The message 233 can be received from MNO 102 with a SUCI 101b, and the symmetric ciphering key 225b selected for decryption from the network database 103d using the SUCI 101b.

The received device certificate 101c decrypted from a message 233 using the symmetric ciphering key 225b can be compared with the device certificate 101c stored in a database for SUCI 101b using a step 224. Network 103 can derive an anchor key 236a for a device 101 with a SUCI 101b using at least the shared secret 225a and the EMSK 236b stored in a network database 103d for the SUCI 101b. Additional examples for the use of data in a network database 103d were provided in Figure 2 above, and other possibilities exist as well for a network 102 and/or an AUSF 102b to store and use data from a network database 103d without departing from the scope of the present disclosure.

Conclusion

Various exemplary embodiments have been described above. Those skilled in the art will understand, however, that changes and modifications may be made to those examples without departing from the scope of the claims.

CLAIMS

1. A method for a server system to support an EAP-TLS authentication with a device, the method performed by the server system, the method comprising:

- a) receiving, from a wireless network, a subscriber concealed identifier (SUCI) and a first device ephemeral public key;
- b) sending, to a device provider, the SUCI and the first device ephemeral public key;
- c) receiving, from the device provider, a subscriber permanent identifier (SUPI), a certificate for the device, and a device provider certificate;
- d) receiving, from the wireless network, a “client hello” message for the device with at least a second device ephemeral public key;
- e) deriving a server ephemeral private key and a corresponding server ephemeral public key;
- f) sending, to the device provider, at least the derived server ephemeral public key;
- g) receiving, from the device provider, a digital signature over at least the derived server ephemeral public key and the device provider certificate, wherein the digital signature is generated by the device provider using at least a device provider private key corresponding to a device provider public key in the device provider certificate;
- h) generating (i) a shared secret key using at least the second device ephemeral public key and the server ephemeral private key and (ii) a shared symmetric ciphering key using at least the shared secret key;
- i) sending, to the wireless network, the SUCI, a “server hello” message with at least the derived server ephemeral public key, the device provider certificate, and the digital signature, wherein at least the device provider certificate and the digital signature in the “server hello” message are encrypted with the shared symmetric ciphering key; and
- j) generating an anchor key for the wireless network using at least the shared secret key, the “client hello” message, and the “server hello” message.

2. The method of claim 1, wherein the “client hello” message and the “server hello” message support a Transport Layer Security version 1.3, and wherein the device generates the “client hello” message.

3. The method of claim 1, wherein the SUCI includes a domain name for the device provider as a first plaintext and a user name for the device as a ciphertext, and wherein the SUPI includes the user name for the device as a second plaintext.

4. The method of claim 1, wherein the “client hello” message includes extensions, and wherein the extensions include cryptographic parameters, and wherein the server ephemeral private key, the corresponding server ephemeral public key, and the second device ephemeral public key support the cryptographic parameters.

5. The method of claim 4, further comprising generating, by the server system, the shared secret using an elliptic curve Diffie Hellman (ECDH) key exchange and the cryptographic parameters.

6. The method of claim 1, wherein the server system comprises at least one server and an authentication server function (AUSF).

7. The method of claim 1, wherein device provider records a secret key, and wherein the device provider decrypts at least a portion of the SUCI using the secret key and the first device ephemeral public key in order to determine the SUPI for the device.

8. The method of claim 1, wherein the device stores the device provider certificate before the device sends the SUCI to the wireless network, and wherein the device verifies the digital signature using at least the device provider public key in the device provider certificate.

9. The method of claim 1, wherein the wireless network encrypts data for the device using at least the anchor key.

10. The method of claim 1, further comprising generating the shared symmetric ciphering key using at least the shared secret key and a key derivation function with the shared secret key, and wherein the key derivation function uses a secure hash algorithm.

11. The method of claim 1, further comprising, after step i) and before step j), receiving from the device a “client finished” message with a certificate verify for the device, wherein the server system verifies the certificate verify using the device certificate.

12. The method of claim 11, further comprising generating the anchor key for the wireless network using at least the shared secret key, the “client hello” message, the “server hello” message, and the “client finished” message.

13. The method of claim 1, wherein the server system sends the wireless network the anchor key through a secure session established between the server system and the wireless network.

14. The method of claim 1, wherein the SUCI includes a plaintext realm for the device provider, and wherein the server system uses the plaintext realm to send the SUCI and the first device ephemeral public key to the device provider.

15. A method for a server system to support an EAP-TLS authentication with a device, the method performed by the server system, the method comprising:

- a) storing a static private key and cryptographic parameters, wherein the device stores a static public key corresponding to the static private key and the cryptographic parameters;
- b) receiving, from a wireless network, a subscriber concealed identifier (SUCI) and a first device ephemeral public key for the cryptographic parameters, wherein the SUCI includes a realm for a device provider as a first plaintext;
- c) decrypting the SUCI using at least the static private key and the first device ephemeral public key in order to read a subscriber permanent identifier (SUPI);
- d) sending, to a device provider at the realm, at least a user name for the SUPI;
- e) receiving, from the device provider, a certificate for the device and a device provider certificate;
- f) receiving, from the wireless network, a “client hello” message for the device with at least a second device ephemeral public key;
- g) deriving a server ephemeral private key and a corresponding server ephemeral public key;
- h) sending, to the device provider, at least the derived server ephemeral public key;
- i) receiving, from the device provider, a digital signature over at least the derived server ephemeral public key and the device provider certificate, wherein the digital signature is generated by the device provider using at least a device provider private key corresponding to a device provider public key in the device provider certificate;
- j) generating (i) a shared secret key using at least the second device ephemeral public key and the server ephemeral private key and (ii) a shared symmetric ciphering key using at least the shared secret key;
- k) sending, to the wireless network, the SUCI, a “server hello” message with at least the derived server ephemeral public key, the device provider certificate, and the digital signature, wherein at least the device provider certificate and the digital signature in the “server hello” message are encrypted with the shared symmetric ciphering key; and
- l) generating an anchor key for the wireless network using at least the shared secret key, the “client hello” message, and the “server hello” message.

16. The method of claim 15, wherein the “client hello” message and the “server hello” message support a Transport Layer Security version 1.3, and wherein the device generates the “client hello” message.

17. The method of claim 15, wherein the SUCI includes a user name for the device as a ciphertext, and wherein the SUPI includes the user name for the device as a second plaintext.

18. The method of claim 15, wherein the “client hello” message includes extensions, and wherein the extensions include cryptographic parameters, and wherein the server ephemeral private key, the corresponding server ephemeral public key, and the second device ephemeral public key support the cryptographic parameters.

19. The method of claim 18, further comprising generating, by the server system, the shared secret using an elliptic curve Diffie Hellman (ECDH) key exchange and the cryptographic parameters.

20. The method of claim 15, wherein the server system comprises at least one server and an authentication server function (AUSF).

21. The method of claim 15, wherein device provider stores a device database for a plurality of devices, and wherein the device provider queries the device database with the user name in order to send the server system the certificate for the device and the device provider certificate.

22. The method of claim 15, wherein the device stores the device provider certificate before the device sends the SUCI to the wireless network, and wherein the device verifies the digital signature using at least the device provider public key in the device provider certificate.

23. The method of claim 15, wherein the wireless network encrypts data for the device using at least the anchor key.

24. The method of claim 15, further comprising generating the shared symmetric ciphering key using at least the shared secret key and a key derivation function with the shared secret key, and wherein the key derivation function uses a secure hash algorithm.

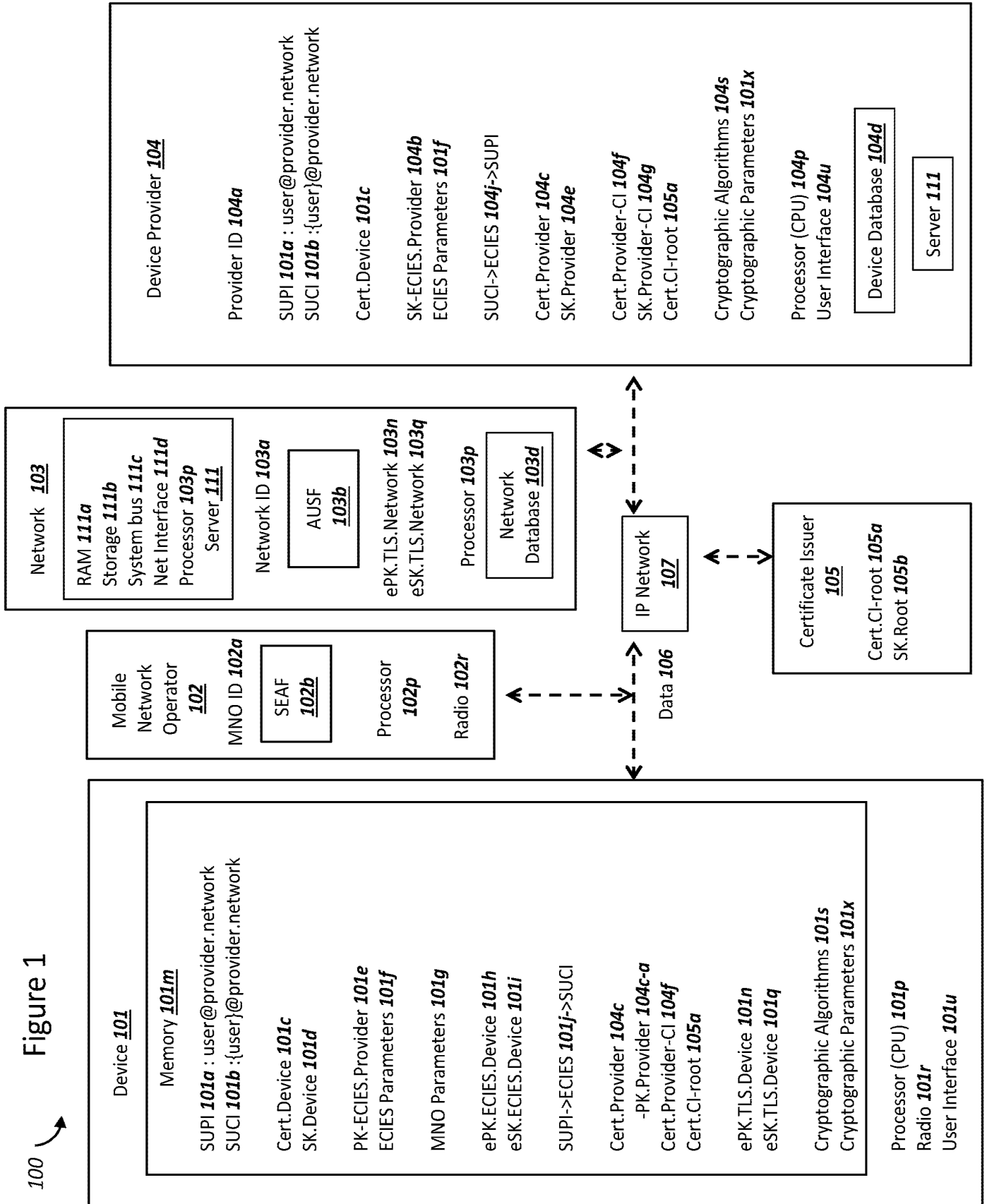
25. The method of claim 15, further comprising, after step i) and before step j), receiving from the device a “client finished” message with a certificate verify for the device, wherein the server system verifies the certificate verify using the device certificate.

26. The method of claim 25, further comprising generating the anchor key for the wireless network using at least the shared secret key, the “client hello” message, the “server hello” message, and the “client finished” message.

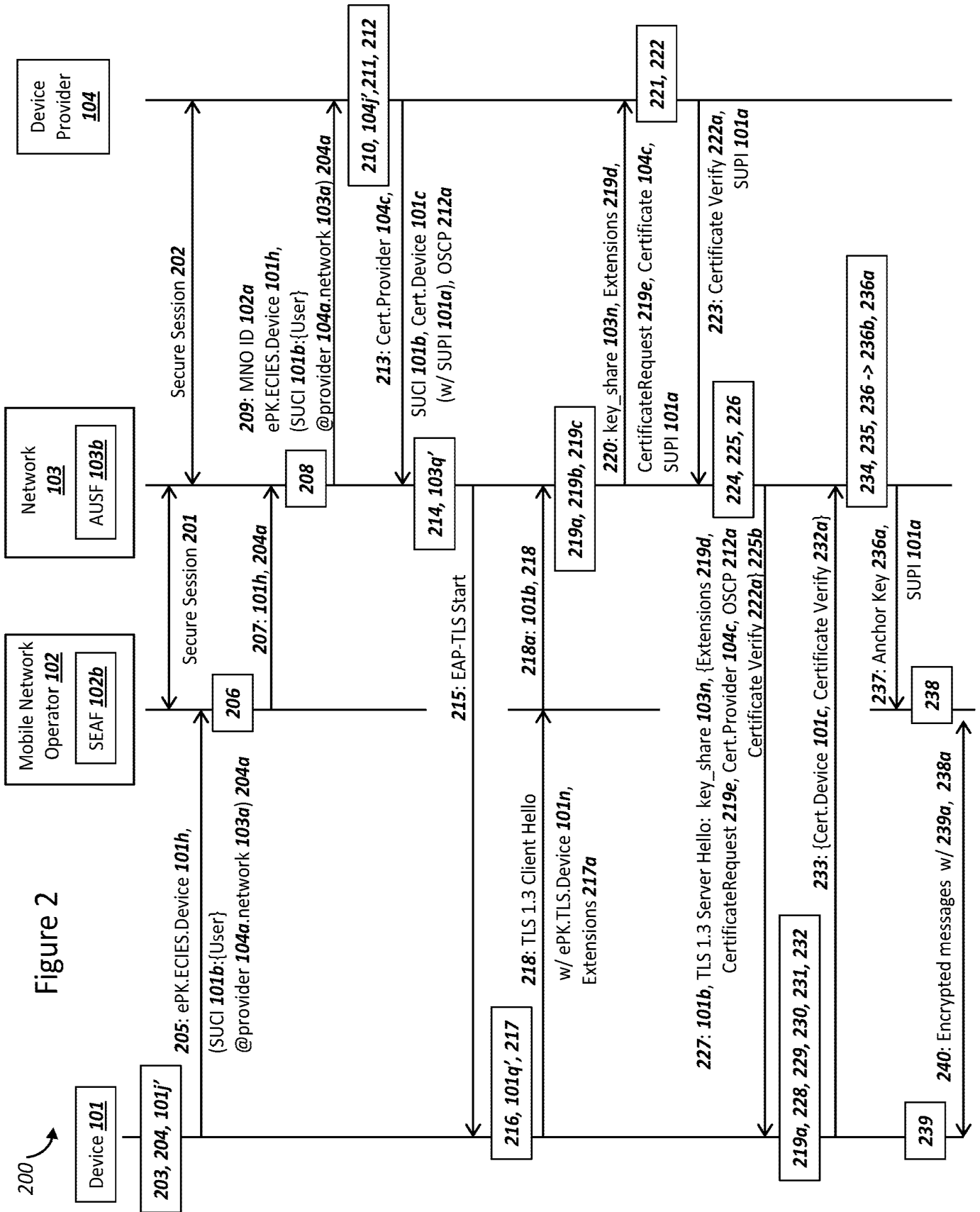
27. The method of claim 15, wherein the server system sends the wireless network the anchor key through a secure session established between the server system and the wireless network.

28. The method of claim 15, wherein the device stores an first device ephemeral private key for the first device ephemeral public key, and wherein the device encrypts the SUPI using the static public key and the first device ephemeral private key in order to generate the SUCI, and wherein the device sends the SUCI to the wireless network.

1/3



2/3



3/3

Figure 3

Device Database 104d									
Device	NAI User		Device ID /		Device	Provider	Provider	Provider	ECIES Secret Key
	Name	SUPI	Certificate	ID	Certificate	Secret Key	ECIES Secret Key	ECIES Parameters	
		101a	101c	104a	104c	104e	104b	101f	• • •
1	user-1	101a-1	101c-1	104a-1	104c-1	104e-1	104b-1	101f-1	
2	user-2	101a-2	101c-2	104a-1	104c-1	104e-1	104b-1	101f-1	
3	user-3	101a-3	101c-3	104a-1	104c-2	104e-2	104b-1	101f-1	
4	user-4	101a-4	101c-4	104a-2	104c-2	104e-2	104b-2	101f-2	
5	user-5	101a-5	101c-5	104a-2	104c-3	104e-3	104b-2	101f-2	
6	user-6	101a-6	101c-6	104a-2	104c-3	104e-3	104b-2	101f-2	
• • •									
Network Database 103d									
Device ID / SUPI	Device ID /		Device	Provider	TLS Server	TLS Device	Shared	Symmetric	Anchor Key
	SUPI	Certificate	Certificate	Ephemeral SK	Ephemeral PK	Secret	key	EMSK	
101b	101a	101c	104c	103q	101n	225a	225b	236b	236a
101b-1	101a-1	101c-1	104c-1	103q-1	101n-1	225a-1	225b-1	206b-1	206a-1
101b-2	101a-2	101c-2	104c-1	103q-2	101n-2	225a-2	225b-2	206b-2	206a-2
101b-3	101a-3	101c-3	104c-2	103q-3	101n-3	225a-3	225b-3	206b-3	206a-3
101b-4	101a-4	101c-4	104c-2	103q-4	101n-4	225a-4	225b-4	206b-4	206a-4
101b-5	101a-5	101c-5	104c-3	103q-5	101n-5	225a-5	225b-5	206b-5	206a-5
101b-6	101a-6	101c-6	104c-3	103q-6	101n-6	225a-6	225b-6	206b-6	206a-6
• • •									

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2020/030424

A. CLASSIFICATION OF SUBJECT MATTER

H04W 12/06(2009.01)**H04W 12/04**(2009.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04W 12/00-12/12, H04W80/00, H04W84/00, H04L9/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Espacenet, PatSearch, PAJ, WIPO, USPTO

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2019/020439 A1 (TELEFONAKTIEBOLAGET LM ERICSSON) 31.01.2019, abstract, [0016], [0017]-[0020], [0046], [0048], [0049], [0076], [0081]), fig. 3, 6	1-28
Y	WO 2018/208228A2 (HUAWEI INT PTE LTD) 15.11.2018, [0137], [0336]-[0363], fig. 15	1-28
Y	5G; Security architecture and procedures for 5G System (3GPP TS 33.501 version 15.1.0 Release 15) [online]. July 2018 [retrieved 04.07.2020]. Retrieved from < https://www.etsi.org/deliver/etsi_ts/133500_133599/133501/15.01.00_60/ts_133501v150100p.pdf >, chapter 6.1.1	15-28



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“D” document cited by the applicant in the international application

“E” earlier document but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T”

later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X”

document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y”

document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&”

document member of the same patent family

Date of the actual completion of the international search

07 July 2020 (07.07.2020)

Date of mailing of the international search report

23 July 2020 (23.07.2020)

Name and mailing address of the ISA/RU:

Federal Institute of Industrial Property,
Berezhkovskaya nab., 30-1, Moscow, G-59,
GSP-3, Russia, 125993
Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37

Authorized officer

E. Krivtsova

Telephone No. 8(499)240-25-91

Form PCT/ISA/210 (second sheet) (July 2019)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2020/030424

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	Badra M. Key-Exchange Authentication Using Shared Secrets. Computer, 39(3) [online]. March 2006, [retrieved 04.07.2020]. Retrieved from < https://www.researchgate.net/publication/2956506_Key-Exchange_Authentication_Using_Shared_Secrets >, pages 58-66, chapters «Certificate issues in wireless networks», «Transport Layer Security-Key-Exchange Method Benefits», «The Transport Layer Security Key-Exchange Method (TLS-KEM) extension uses an out-ofband shared secret to integrate TLS with wired and wireless networks without influencing security robustness», «Discussion and results»	15-28
Y	E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3 [online]. August 2018 [retrieved 04.07.2020]. Retrieved from < https://tools.ietf.org/html/8446 >, chapters 2, 4.1.2, 4.1.3, 4.1.6, 4.2.3, 4.2.4, 4.4.1, 4.4.2, 7.3 7.4.2, 8.1	2, 4, 5, 8-13, 16, 18-27
A	US 7774411 B2 (WISYS TECHNOLOGY FOUNDATION, INC.) 10.08.2010	1-28

Form PCT/ISA/210 (continuation of second sheet) (July 2019)

REQUEST FOR CONTINUED EXAMINATION(RCE)TRANSMITTAL (Submitted Only via EFS-Web)

Application Number	16/453,682	Filing Date	2019-06-26	Docket Number (if applicable)	63121/0184	Art Unit	2432
First Named Inventor	John A. Nix			Examiner Name	HO, Dao Q.		

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.

Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. The Instruction Sheet for this form is located at WWW.USPTO.GOV

SUBMISSION REQUIRED UNDER 37 CFR 1.114

Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in the order in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).

- ☐ Previously submitted. If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.
- ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on _____
- ☐ Other _____
- ☒ Enclosed
- ☐ Amendment/Reply
- ☒ Information Disclosure Statement (IDS)
- ☐ Affidavit(s)/ Declaration(s)
- ☐ Other _____

MISCELLANEOUS

- ☐ Suspension of action on the above-identified application is requested under 37 CFR 1.103(c) for a period of months _____
(Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)
- ☐ Other _____

FEES

- ☒ The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.
- ☒ The Director is hereby authorized to charge any underpayment of fees, or credit any overpayments, to Deposit Account No 011785

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

- ☒ Patent Practitioner Signature
- Applicant Signature

Signature of Registered U.S. Patent Practitioner			
Signature	Albert J. Boardman/	Date (YYYY-MM-DD)	2021-08-05
Name	Albert J. Boardman	Registration Number	70601

This collection of information is required by 37 CFR 1.114. The information is required to obtain or retain a benefit by the public which is a file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

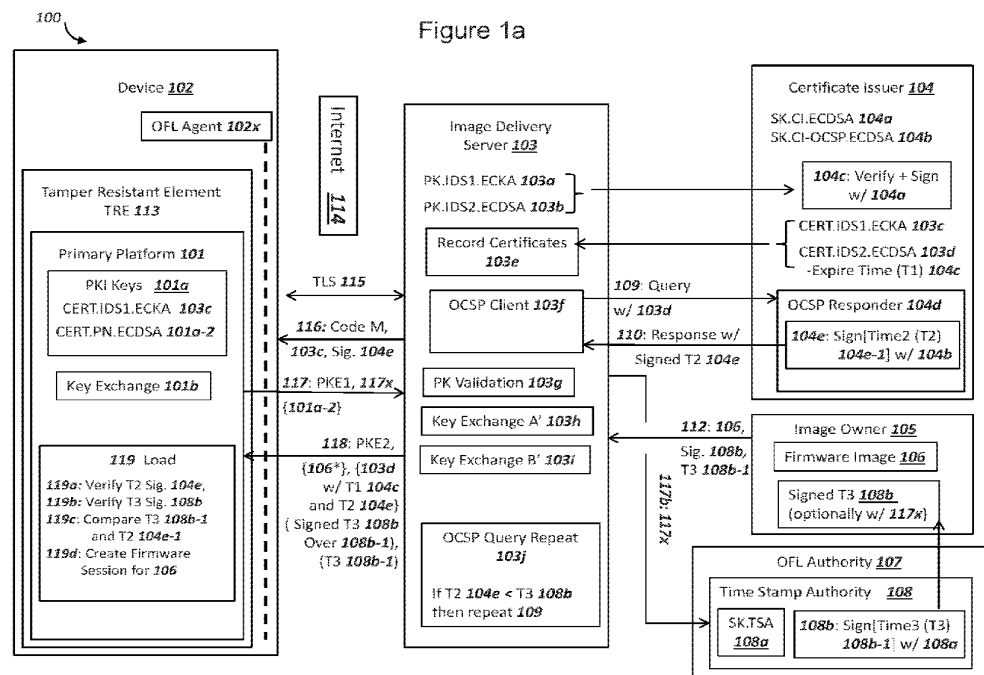
The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.



- (51) **International Patent Classification:**
G06F 21/64 (2013.01) *H04L 9/14* (2006.01)
- (21) **International Application Number:**
PCT/US2019/066257
- (22) **International Filing Date:**
13 December 2019 (13.12.2019)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
62/779,791 14 December 2018 (14.12.2018) US
- (71) **Applicant:** IOT AND M2M TECHNOLOGIES, LLC
[US/US]; 807 Davis Street, Unit 2207, Evanston, IL
60201-7104 (US).
- (72) **Inventor:** NIX, John, A.; c/o IoT and M2M Technologies, LLC, 807 Davis Street, Unit 2207, Evanston, IL
60201-7104 (US).
- (74) **Agent:** THOMSON, Kirsten, L.; McDonnell Boehnen
Hulbert & Berghoff LLP, 300 South Wacker Drive, Chicago,
IL 60606 (US).
- (81) **Designated States** (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,
KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME,
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,
OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,
SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(54) **Title:** SECURE IDS CERTIFICATE VERIFICATION FOR A PRIMARY PLATFORM

(57) **Abstract:** A tamper resistant element (TRE) in a device can operate a primary platform and support a "Smart Secure Platform". The TRE may not keep time when electrical power is removed from the TRE. The device can receive (i) a certificate for an image delivery server (IDS) with a first timestamp and (ii) a signed second timestamp from a certificate authority, comprising a signature according to the Online Certificate Status Protocol (OCSP) with stapling. The device can forward the certificate and second timestamp to the TRE. The device can receive a ciphertext and an encrypted image from the IDS, where the ciphertext includes a third timestamp from a Time Stamp Authority (TSA), and forward the data to the TRE. The TRE can conduct a key exchange to decrypt the ciphertext. The TRE can compare the second and third timestamps to verify the certificate has not been revoked.

[Continued on next page]

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

SECURE IDS CERTIFICATE VERIFICATION FOR A PRIMARY PLATFORM**CROSS-REFERENCE TO RELATED APPLICATION**

This international PCT application claims the benefit of the filing date of U.S. Provisional Patent Application Serial No. 62/779,791, filed December 14, 2018, which is hereby incorporated by
5 reference in its entirety.

BACKGROUND**Technical Field**

The present systems and methods relate to securely transferring firmware to a primary
10 platform from a server, and more particularly for a tamper resistant element (TRE) to determine if a certificate or public key associated with an image delivery server has been revoked.

Description of Related Art

The commercial development and deployment of secure processing environments within
15 microprocessors and tamper resistant elements (TRE) can increase security of computing devices such as mobile phones and networked sensors for the “Internet of Things”. A secure processing environment or TRE can enable a device to securely record cryptographic keys and conduct cryptographic operations such as key exchanges, key derivation, and also digital signatures in a computing environment that is isolated from other electrical components within the device that may
20 not be secure. Examples available today include an embedded subscriber identity module (embedded SIM), which is also known as an embedded universal integrated circuit card (eUICC), a traditional SIM card, a secure enclave within a “system on a chip”, a trusted execution environment (TEE), and other examples are available as well. A common computing architecture includes a processor with multiple cores, where a secure processing core is isolated from the other cores, and
25 the secure processing core can read cryptographic keys and conduct cryptographic operations. Other possibilities exist as well, including “switching” a generic processor between an insecure mode to a secure mode.

As the number of transistors and memory cells available for a given surface area of silicon continues to grow, the computational power for secure processing environments continues to
30 increase. Some secure processing environments can operate as a host computing environment and provide the functionality of a virtual machine for different firmware images, such that each firmware

image can operate as separate computing environments within the secure processing environment. As one example, secure processing environments can now enable a TRE to operate as a primary platform for hosting potentially a variety of different firmware images, where each of the firmware images can support a different application. As one example, a secure processing environment could
5 operate a Java™ virtual machine, and different firmware images could comprise different Java applets or Java-based applications. Each Java applet could comprise different images for the Java virtual machine. A firmware can also comprise the base operating system for a primary platform, such as a primary firmware of machine executable program instructions for the primary platform to operate as a host for secondary firmware to operate as virtual machines. Other possibilities exist as
10 well for a secure processing environment to operate as a primary platform for hosting images, and the images may also be referred to as a firmware image.

A primary platform both (i) operating within a secure processing environment and (ii) using firmware images can support a variety of useful applications for a computing device. The operation of an example primary platform can perform functions outlined in “Open Firmware Loader for
15 Tamper Resistant Element”, version 1.3 from Global Platform dated June 2017, which is hereby incorporated by reference in its entirety (“GP OFL document”). An open firmware loader can be referred to as “OFL”. The operation of an example primary platform can also perform functions outlined in the GSM Association (GSMA) technical document “iUICC POC Group Primary Platform requirements”, Release 1.0 dated May 17, 2017, which is hereby incorporated by reference
20 in its entirety (“GSMA PP Requirements”). Example applications supported by a primary platform using firmware images are identified in Section 3 “Use Cases” of the GSMA PP Requirement. Example applications include firmware for an embedded SIM, a traditional universal integrated circuit card (UICC), mobile payments, secure bootstrapping, digital rights management, user identification such as a drivers license, secure access to home automation, a virtual automobile key,
25 and other applications are identified as well.

Further the European Telecommunications Standards Institute (ETSI) has begun developing standards for a “Secure Primary Platform” (SSP) as part of the development of 5G standards, and an SSP could operate as a primary platform as well. As of December, 2018, the draft standards for an SSP are not available for public review, but will likely support applications similar to those
30 contemplated in the GSMA PP requirements document and GP OFL document. As the standards for operating a primary platform or a secure primary platform (SSP) continue to evolve, new features or the use of new cryptographic algorithms or steps may be introduced. These new features and/or cryptographic steps may require new firmware for the primary platform in order to support new or updated versions of the standards. A need exists in the art for a primary platform to securely receive
35 updated firmware in order to support currently evolving standards and features for the operation of a primary platform or SSP and supported applications.

Secure operation of a primary platform or an SSP for each of the above standards depends on the secure delivery of firmware from a server to the primary platform. Different firmware may be required by a primary platform in order to support each of the above example applications. Further, over time new features may be added to the applications, which would also likely require a firmware update. The GP OFL document provides an overview of a proposed solution in Figure 2-1 on page 15. Details for an OFL security scheme are outlined in the GP OFL document in Figure 4-1 on page 33. Secure encryption and also an authenticated delivery of firmware to a primary platform is a significant technical challenge, since the computing device may include insecure components, such as a generic processor and a generic operating system. The insecure device can conduct many of the steps for communicating between (i) a server that sources the firmware and (ii) the primary platform. As one example, the computing device could comprise a mobile phone or “smartphone” based on Android or IOS or similar operating systems and could also either (i) operate with “malware” that is unknown to a user or a network or (ii) could comprise a “rooted” mobile phone that is under the control of hackers. Many other examples of insecure devices operating a TRE with a primary platform exist as well. Note that the GP OFL document does not include the words “revocation” or “revoked”, which are associated with steps to revoke or invalidate a previously issued certificate from a certificate issuer. A need exists in the art for secure steps for a TRE with a primary platform to determine revocation of a certificate or public key for an image delivery server (IDS), an OFL, or an associated server.

A primary platform or a “Smart Secure Platform” operating in a tamper resistant element can be a resource constrained computing environment, compared to a traditional computing environment of a mobile phone or a personal computer. This resource constrained environment can create significant challenges for revocation of a certificate or public key associated with a server supporting the delivery of firmware to the primary platform. As one example, a primary platform may not have an internal clock to keep track of local date and time when power is removed from the primary platform. The primary platform must communicate through the insecure mobile phone or computing device in order to send and receive data with a server, including updated firmware for the primary platform. The primary platform may not have resources to operate a full transport layer security (TLS) stack. In addition, the primary platform may lack computing resources and electrical components for the proper and full implementation of certificate revocation lists, the Online Certificate Status Protocol (OCSP), or associated schemes for checking the potential revocation of a certificate or public key from a server. A need exists in the art for a primary platform operating in a tamper resistant element to use time values received in order to securely determine if a certificate or public key used with cryptographic operations has been revoked.

Both (i) the long timeframes for a tamper resistant element to record and operate with server certificates, which could comprise certificates structured according to the X.509 family of standards

and include a certificate issuer (CI) root certificate, and (ii) the use of potentially hundreds or more of IDS certificates signed by the CI root private key increases the importance and likelihood of a need in the art for a TRE to support or determine certificate revocation for a certificate received by the TRE. As one example, the root certificate from the GSMA for the system supporting the embedded universal integrated circuit card (eUICC) currently expires in approximately 2052. The CI certificate recorded in devices for a system of TRE as contemplated by GP OFL may similarly expire in future decades. An IDS may communicate with millions of devices with TREs and hundreds of service providers over time, and may prefer to use an IDS certificate that is valid for several years for operational purposes.

The GP OFL contemplates in section 4.2.6 on page 39 checking for validity of an IDS certificate using a timestamp received, but note this only contemplates the verification that an IDS certificate has not expired, where expiration of the IDS certificate may be several years into the future. There is no teaching or suggestion in the GP OFL for an intermediate check or verification by the TRE that if the IDS certificate or OFL authority certificate has been revoked. A need exists in the art for a primary platform in a TRE to conduct a check for certificate revocation where the device operating the TRE may not directly or efficiently or reliably communicate with the certificate issuer (CI) that signed a certificate used by the TRE, such as an IDS certificate or an image owner certificate.

Note that the IDS may be connected to the public Internet in order to support the potential millions of different devices with a TRE requesting firmware for the PP. This creates potential security risks for protecting the private key of the IDS corresponding to the public key stored by the TRE with an IDS certificate, especially over long period of time such as several years. As one example, the “Heartbleed” vulnerability in OpenSSL potentially exposed the private key operated by servers connected to the Internet. The cloud hosting service Cloudflare around 2014 had to revoke and reissue thousands of server certificates for that one vulnerability. Another, similar vulnerability may be found in the future, and several existing threats for server security continue in the families of Spectre and Meltdown vulnerabilities. The list from Common Vulnerabilities and Exposures contains thousands of entries for the year 2018 alone. Many of these vulnerabilities could result in exposure or leaking of bits of security for a private key of a server corresponding to the public key in a certificate operated by a TRE as contemplated by Global Platform and ESTI standards. Consequently, a need exists in the art for a certificate issuer to revoke a certificate used by a primary platform or “Smart Secure Platform” and reissue the certificate in a manner that the TRE can determine that the certificate has been revoked.

Many other examples exist as well for needs in the art for a primary platform to securely receive firmware from a server using certificates where the primary platform can determine if the

certificate has been revoked, and the above are examples are just a few and intended to be illustrative instead of limiting.

SUMMARY

Methods and systems are provided for a primary platform to securely receive firmware from a server. A device can include a tamper resistant element. The primary platform (PP) can operate within the tamper resistant element (TRE) and comprise a secure processing environment operating within a computing device. The primary platform can comprise a secure enclave, a secure element, a trusted execution environment (TEE), or a “Smart Secure Platform” as specified by ETSI standards. The primary platform can also operate within an ARM® Trustzone implementation of a TEE. The TEE can support GlobalPlatform standards for a TEE. The computing device can comprise a mobile phone, wireless device, personal computer, laptop, or a networked transducer for the “Internet of Things”. The device can (i) include insecure components such as a general processor and a general operating system and (ii) communicate with the primary platform using a device driver such as an Open Firmware Loader (OFL) agent.

The device can connect with an internet protocol (IP) based network such as the public Internet in order to establish a secure session with a server. The server can comprise an image delivery server (IDS) and receive a firmware image from an image owner. The server can comprise a computer with a network interface to communicate with the device via the IP network. The server can record and operate a server database. The device can be one of a plurality of different devices communicating with the server. The server can operate an online certificate status protocol (OCSP) client in order to periodically obtain a signed verification of certificate validity from a certificate issuer.

In exemplary embodiments an image owner can obtain a firmware image for the primary platform. The firmware image can support the primary platform operating with an application, such as, but not limited to, an eUICC, a “Smart Secure Platform”, secure identification of the device using the primary platform, and other possibilities exist as well for supported applications. The image owner can obtain the firmware image for a plurality of different primary platforms. The firmware image from the image owner can comprise an unbound image, which could be subsequently used by the server communicating with a plurality of different primary platforms, after subsequent cryptographic operations by the server with the plurality of different primary platforms in order to create bound images. The image owner can receive a signed timestamp from an open firmware loader (OFL) authority, where the OFL authority also provides additional authorization information for the primary platform. The image owner can send the authorization information and the signed timestamp to the server. For some embodiments, the use of a primary platform could be omitted, and the firmware image could comprise firmware for a processor within a device operating in a secure manner (e.g. Trustzone, TEE, “secure enclave”, etc.). In addition, the image owner or IDS

could receive the signed timestamp from a server different than a server associated with the OFL authority.

The primary platform and the server can record and operate a set of compatible cryptographic parameters and cryptographic algorithms for an elliptic curve Diffie Hellman (ECDH) key exchange algorithm, a symmetric ciphering algorithm, and elliptic curve digital signature algorithms (ECDSA). Before distribution to an end user of the computing device, a TRE manufacturer could record a set of data in nonvolatile memory for the TRE. In addition to a PP boot firmware and PP boot configuration for the PP, the data recorded in device before distribution could include a root certificate for a certificate issuer and a certificate for a timestamp authority. Other PKI keys could be recorded by a TRE manufacturer in a set of PKI keys for the primary platform as well

A certificate issuer (CI) can receive a first public key and a second public key from the server and issue certificates for each of the public keys. The certificate issuer can use a first CI private key, corresponding to a CI root certificate recorded by the primary platform, to sign the server public keys. The first public key can be associated with elliptic curve key agreement and the second public key can be associated with elliptic curve digital signature algorithms (ECDSA). The certificate issuer can also (i) operate an OCSP responder to receive queries for interim certificate validity before the expiration time of the certificates and (ii) respond with a signed response and a time value, where the time value represents the end of an interim time into the future that the certificate can be considered valid and not revoked. In exemplary embodiments, the interim time could comprise weekly updates for the time value and interim verification for the validity of the certificate. The responses from the OCSP responder can use a second CI private key, where either (i) a certificate for the second CI private key is also recorded by the primary platform, or (ii) a certificate for the second CI private key is signed by the first CI private key and the certificate is sent along with the response for the interim update for certificate validity.

After boot and startup of both the device and the TRE operating the primary platform, the device can use an OFL agent as a device driver to communicate with the primary platform. The device can use the OFL agent to obtain (i) a firmware status, (ii) the primary platform identity, and (iii) a certificate for the primary platform. The firmware status can comprise a list of parameters associated with any firmware image recorded or stored by the primary platform, including a version number for either a stored firmware image or a version number for PP boot firmware. In exemplary embodiments, the set of cryptographic parameters in the certificate for the primary platform at least specifies a specific ECC curve for use with the primary platform private key. The device can establish a secure session with the server and receive a first server certificate for the server, where the certificate includes OCSP stapling of a current OCSP response from the certificate issuer for the first server certificate. The OCSP response with the first server certificate can include a first time

value. The device can forward the first server certificate with the OCSP response and the first time value to the primary platform using the OFL agent.

The primary platform can receive the first server certificate with the OCSP response and first the time value. The primary platform can use a certificate for the certificate issuer to verify the first time value, where the first time value is stored for a later comparison with a received timestamp. The primary platform can derive an ephemeral PKI key pair comprising a first ephemeral private key and a first ephemeral public key, using parameters from the first server certificate. The primary platform can conduct a first key exchange in order to derive a first symmetric ciphering key and encrypt the certificate for the primary platform using the derived first symmetric ciphering key. The primary platform can send the ephemeral public key and the encrypted certificate for the primary platform to the OFL agent in the device, which forwards the data to the server through the secure connection.

The server can receive the ephemeral public key and the encrypted certificate for the primary platform and conduct a public key validation step for the ephemeral public key. The server can conduct a second key exchange using the received first ephemeral public key and a first server private key corresponding to the server public key sent in the first server certificate, in order to derive the first symmetric ciphering key. The server can decrypt the primary platform certificate using the derived first symmetric ciphering key.

The server can derive an ephemeral PKI key pair which includes a second ephemeral public key and a second ephemeral private key. The server can conduct a series of steps to encrypt the received firmware image and the signed timestamp, where the series of steps include using (i) the public key from the primary platform certificate, (ii) a second private key corresponding to a second public key in a second server certificate, and (iii) the derived second ephemeral private key. The server can send a message to the device with at least the second ephemeral public key, the encrypted image, the signed timestamp, and the second server certificate. The message with the second server certificate can also include OCSP stapling of a current OCSP response from the certificate issuer for the certificate. The OCSP response with the second certificate can include a second time value.

The device can receive the message from the server and use the OFL agent to send the second ephemeral public key, the encrypted image, the signed timestamp, and the second server certificate to the primary platform operating in the tamper resistant element. The primary platform can conduct a series of steps in a firmware load function in order to process the data received. The device can use (a) the received second ephemeral public key and a private key corresponding to the public key in the primary platform certificate to (b) derive at least a second symmetric ciphering key. The primary platform can use the derived symmetric ciphering key and a symmetric ciphering algorithm to decrypt the encrypted data forwarded from the device in the message to the TRE. The primary platform can decrypt the encrypted firmware image. The message can include an access

rights pattern that is signed by the image owner, and the primary platform can verify the signature for the access rights platform using a public key for the image owner. The primary platform can use the access rights pattern to determine if loading and operating with the firmware image is authorized, as well as other constraints for the firmware image.

5 The primary platform can verify the signed timestamp using a recorded certificate for the timestamp authority. The primary platform can use a certificate for the certificate issuer to verify a second time value from the OCSP stapling with the second server certificate. The primary platform can compare both (i) the first time value from OCSP stapling for the first server certificate and (ii) the second time value from OCSP stapling for the second server certificate with (iii) the received,
10 verified timestamp value. If the received, verified timestamp value is before both the first time value and the second time value (where the first time value and the second time value can represent the end time for validity of OCSP stapling values), then the primary platform can continue the firmware load function and establish a firmware session using the received, decrypted, and authenticated firmware from the message.

15 If the received, verified timestamp value is after either the first time value or the second time value, then the primary platform cannot trust or rely that both the first server certificate and the second server certificate remain valid and have not been revoked. If the received, verified timestamp value is after either the first time value or the second time value, then the primary platform can send an error response to the server through the device and stop the firmware load
20 process, so that the received firmware is not loaded until certificates with time values (e.g. outside window of time validity or “expiration” of the OCSP stapling data) greater than the timestamp are received and verified by the primary platform.

BRIEF DESCRIPTION OF THE DRAWINGS

Various exemplary embodiments are described herein with reference to the following drawings, wherein like numerals denote like entities.

5 Figure 1a is a graphical illustration of an exemplary system, where a device with a tamper resistant element and a server securely transfer encrypted firmware from the server to the tamper resistant element, in accordance with exemplary embodiments;

10 Figure 1b is a graphical illustration of hardware, firmware, and software components for a device, including a tamper resistant element with a primary platform, in accordance with exemplary embodiments;

 Figure 1c is a graphical illustration of a primary platform, in accordance with exemplary embodiments;

 Figure 2a is a simplified response from a certificate authority to verify certificate validity, in accordance with exemplary embodiments; and,

15 Figure 2b is a flow chart illustrating exemplary steps for verifying a digital signature using PKI keys, parameters, and data input, in accordance with exemplary embodiments.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS OF THE INVENTION**Figure 1a**

Figure 1a is a graphical illustration of an exemplary system, where a device with a tamper resistant element and a server securely transfer encrypted firmware from the server to the tamper resistant element, in accordance with exemplary embodiments. The system 100 can include a device 102, a server 103, a certificate issuer 104, an image owner 105, an open firmware loader (OFL) authority 107, and a network 114. The various computing nodes depicted in Figure 1a can establish secure sessions such as a Transport Layer Security (TLS) 115 session over an Internet Protocol (IP) network 114, including the TLS 115 session depicted between device 102 and server 103. Server 103 can comprise an image delivery server (IDS). Although a single server 103, device 102, a CI 104, etc. are depicted in Figure 1a, a system 100 can comprise a plurality of the computing nodes. In addition, although a single server 103 is depicted in Figure 1a, the exemplary server 103 shown for system 100 can comprise either different physical computers such as rack-mounted servers, or different logical or virtual servers or instances operating in a “cloud” configuration, including different computing processes which are geographically dispersed.

Device 102 can be a computing device for sending and receiving data, including a wireless device. Device 102 can take several different embodiments, such as a general purpose personal computer, a mobile phone based on the Android ® from Google ® or the IOS operating system from Apple ®, a tablet, a device with a sensor or actuator for the “Internet of Things”, a module for “machine to machine” communications, a device that connects to a wireless or wired Local Area Network (LAN), and other possibilities exist as well without departing from the scope of the present disclosure. Device 102 can also be a computing device according to GSMA technical document “iUICC POC Group Primary Platform requirements”, Approved Release 1.0 dated May 17, 2017, which is hereby incorporated by reference in its entirety (“GSMA PP Requirements”). Device 102 can comprise a device such as that depicted in Figure 6 on page 24 of the GSMA PP Requirements. Exemplary electrical components within a device 102 are depicted and described in Figure 1b below. Other possibilities exist as well for the physical embodiment of device 102 without departing from the scope of the present disclosure.

Server 103 could also represent different logical “server-side” processes within a network, including different programs running on a server that listen and communicate using different IP port numbers within one physical server. In exemplary embodiments, server 103 can operate using the physical electrical components similar to those depicted and described for a device 102 in Figure 1b below. Server 103 can use electrical components with larger capacities and larger overall power consumption, compared to the capacity and power consumption for the equivalent electrical

components in a device 102. Other possibilities exist as well for the physical embodiment of server 103 without departing from the scope of the present disclosure.

IP network 114 could be either a Local Area Network (LAN) or a Wide Area Network (WAN), or potentially a combination of both. IP network 114 could include data links supporting either IEEE 802.11 (WiFi) standards. Device 102 could also utilize a variety of WAN wireless technologies to communicate data in a secure session 115 with server 103, including Low Power Wide Area (LPWA) technology, 3rd Generation Partnership Project (3GPP) technology such as, but not limited to, 3G, 4G Long-Term Evolution (LTE), or 4G LTE Advanced, NarrowBand – Internet of Things (NB-IoT), LTE Cat M, planned future 5G networks, and other examples exist as well. Server 103 can connect to the IP network 114 via a wired connection such as, but not limited to, an Ethernet, a fiber optic, or a Universal Serial Bus (USB) connection (not shown). IP network 114 could also be a public or private network supporting Internet Engineering Task Force (IETF) standards such as, but not limited to, such as, RFC 786 (User Datagram Protocol), RFC 793 (Transmission Control Protocol), and related protocols including IPv6 or IPv4. A public IP network 114 could utilize globally routable IP addresses and also comprise an insecure network.

As depicted in Figure 1a, device 102 can include an OFL agent 102x and a tamper resistant element 113. A device 102 can also include other electrical components and data as depicted and described in connection with Figure 1b below, where the components for a device 102 are described in more detail. An OFL agent 102x can comprise a device program such as a device driver operating on a general processor outside of the tamper resistant element 113. Tamper resistant element (TRE) 113 can comprise a tamper resistant element as described in the GSMA PP Requirements document. Tamper resistant element can comprise a silicon enclave within a tamper resistant chip such as a “system on chip” as depicted and described in connection with Figure 1b below.

TRE 113 in device 102 can include a primary platform (PP) 101, where a primary platform is also described in the GSMA PP Requirements document and the GP OFL document. TRE 113 could also comprise a “Smart Secure Platform” (SSP) as described in ETSI TC SCP Meeting #81 document “SCP(17)000188”, which is hereby incorporated by reference in its entirety. Note that draft specifications for an SSP such as “103 666-1 SSP Draft Specification 0.8.0” are not publicly available and have restricted access on the ETSI web site as of December 15, 2018. Primary platform 101 can comprise a secure operating environment, a secure enclave, a secure element, and include a dedicated processing core within a processor for device 102. Primary platform 101 can also operate in a Trusted Execution Environment (TEE) within a processor for device 102, or also within an ARM® “Trustzone” processing environment. Primary platform 101 can also comprise a SSP as contemplated by ETSI documents and draft specifications for 5G networks.

TRE 113 and PP 101 can support a variety of applications. TRE 113 can comprise the physical device such as that depicted in Figure 1a and Figure 1b below (e.g. silicon, electrical

contacts, housing, etc.), and a primary platform 101 can comprise a secure processing environment operating within the TRE 113 (e.g. operating system, drivers, etc.). With appropriate firmware 106, TRE 113 and PP 101 could operate as an “integrated universal integrated circuit card” (iUICC), an “embedded universal integrated circuit card” (eUICC), a secure element for banking applications or payments from mobile phones, an radio-frequency identity (RFID) card, a secure bootstrap environment for device 102, a virtual key for cars or door locks, an secure environment for recording an identity and secret or private keys for drivers licenses, passports, online or web-site access, etc. Other applications for firmware 106 operating in TRE 113 and PP 101 are possible as well, without departing from the scope of the present disclosure. In general, cryptographic keys and cryptographic algorithms and parameters could be stored in PP 101 in order to securely support applications such as device programs operating on device 102. In this manner, an insecure device program also operating on device 102 would not feasibly be able to ready the cryptographic keys or use the cryptographic algorithms stored in PP 101.

Each of the above exemplary applications can be operated by a firmware 106 running within TRE 113 on PP 101. Although a single firmware 106 is depicted and described in connection with Figure 1a, a TRE 113 and PP 101 can operate a plurality of different firmware 106 simultaneously (or “effectively simultaneously” such that rapid switching of a processor in TRE could make the firmware to appear to operate concurrently over a longer period such as a millisecond, even though only one firmware may operate at one time over a shorter period such as a nanosecond). A primary firmware can comprise the host environment for the primary platform and a secondary firmware can comprise a guest or virtual machine operating in the primary platform. A firmware 106 can comprise either the primary firmware or the secondary firmware as described in the previous sentence and other references herein.

Firmware 106 (as secondary firmware) can operate within a virtual machine for a processor for TRE 113 and PP 101. As one exemplary embodiment, firmware 106 can comprise a JAVA applet running on a java virtual machine in TRE 113, where the JAVA virtual machine can be recorded in secure boot firmware for TRE 113 and PP 101. Different firmware 106 operating within TRE 113 can be isolated from each other by conventional technology for processing hosts and/or virtual machines. Other possibilities exist as well for a TRE 113 and PP 101 to operate as a host for an application downloaded as firmware 106. In summary, the overall security of an application operated by TRE 113 and PP 101 can depend on securely receiving firmware 106 by TRE 113 and PP 101 from a server 103.

PP 101 can include a set of PKI keys 101a, a key exchange algorithm 101b, and a load function 119. Other data, functions, and electrical components can be included in a PP 101, such as the data, functions, and electrical components for a primary platform as described in the GSMA PP requirements document and the GP OFL document. As one example, but not depicted in Figure 1a,

a primary platform 101 can record a globally unique identity for the primary platform and/or TRE 113. A set of PKI keys 101a can record a plurality of certificates and private keys for primary platform 101, which can include a certificate for (i) the IDS server for elliptic curve cryptography key agreement (ECKA) of CERT.IDS1.ECKA 103c and (ii) the chip part number for a elliptic curve digital signature algorithm (ECDSA) of CERT.PN.ECDSA 101a-2. A set of PKI keys 101a can also include the exemplary keys depicted and described in connection with Figure 1c below. CERT.PN.ECDSA 101a-2 along with the corresponding private key can be recorded within PP 101 upon manufacturing of the TRE 113. CERT.IDS.ECKA 103c can be received by PP 101 after manufacturing, such as during setup of a TLS 115 session and before PP 101 conducts the steps for a key exchange algorithm A 101b.

Certificates depicted in Figure 1a such as CERT.IDS.ECKA 101a-1 and CERT.IDS2.ECDSA 103d can specify values or settings for (i) conducting an ECDH or ECDHE key exchange, (ii) mutually deriving a symmetric ciphering key, (iii) an ECC curve which could comprise a commonly used name curve, (iv) PKI key lengths, and/or (iv) an elliptic curve digital signature algorithm (ECDSA), etc. Each of the certificates associated with PP 101, server 103, and other nodes in Figure 1a can record at least one compatible set of cryptographic parameters within the depicted certificates. In exemplary embodiments, the certificates in Figure 1a can specify values for algorithms supporting post-quantum cryptography, such as lattice-based algorithms, code-based algorithms, or multivariate-based algorithms, and other possibilities exist as well without departing from the scope of the present disclosure.

Cryptographic parameters within certificates can specify values for an elliptic curve cryptography (ECC) curve name, key length, key formatting (e.g. compressed or uncompressed), encoding rules, etc. As contemplated herein, the parameters and cryptographic algorithms used with certificates, ECC PKI keys and key exchanges or elliptic curve digital signature algorithms in the present disclosure can be compatible and substantially conform with ECC algorithms and keys as specified in (i) the IETF Request for Comments (RFC) 6090 titled "Fundamental Elliptic Curve Cryptography Algorithms", and (ii) IETF RFC 5915 titled "Elliptic Curve Private Key Structure", and also subsequent and related versions of these standards. In addition, parameters and algorithms for ECDSA in certificates can be compatible and support BSI Technical Guideline TR-03111: Elliptic Curve Cryptography. Version 2.0. Other possibilities exist as well for cryptographic parameters and algorithms specified in a certificate without departing from the scope of the present disclosure.

For use of ECC algorithms, parameters in a certificate can specify elliptic curve names such as, but not limited to NIST P-256, sect283k1, sect283r1, sect409k1, sect409r1, and other possibilities exist as well. Further, elliptic curves that do not depend on curves currently specified by the National Institute of Standards and Technology (NIST) could be utilized as well, such as, but

not limited to, Curve22519, curve448, or FourQ. Parameters in a certificate can specify domain parameters for elements in system 100 to calculate values or numbers in a compatible manner, such as (i) a common base point G for use with ECC PKI key pairs and (ii) a defining equation for an elliptic curve.

5 Key exchange algorithm 101b can use CERT.IDS1.ECKA 103b and a derived ephemeral PKI key pair for PP 101 in order to derive a symmetric ciphering key M1 and message authentication code key H1. The public key for the derived ephemeral PKI key pair can comprise PKE1 depicted for message 117 in Figure 1a. Key exchange algorithm 101b can comprise the steps for section 4.1.2.1 and 4.1.2.2 on page 35 of the GP OFL document. For embodiments that support
10 the use of post-quantum cryptographic algorithms instead of ECC algorithms, the key exchange algorithm 101b can comprise a key exchange mechanism, where the symmetric ciphering key M1 could be encrypted using a public key and decrypted using a private key.

Load function 119 can use data received in message 118, including an ephemeral public key for the IDS server of PKE2 in order to authenticate and decrypt an encrypted firmware 106*. As
15 contemplated herein, an encrypted firmware 106 can be depicted and described herein as “firmware 106*” and a plaintext firmware 106 for a primary platform to establish a firmware session can be depicted and described as “firmware 106”. A load function 119 for a primary platform is also depicted and described in connection with Figure 1c below. Individual steps within a load function 119 in Figure 1a are also described below with the description of message flows between the nodes
20 in system 100 and operation of PP 101.

Firmware 106 recorded in PP 101 upon conclusion of a load function 119 can provide machine executable instructions for a processor in TRE 113 to execute or run. Firmware 106 could comprise a collection of compiled software libraries and programming code for the operation of TRE 113 and PP 101. Firmware 106 could comprise a Java-based applet or application as secondary
25 firmware, where boot firmware of PP 101 establishes and operates a Java virtual machine such as, but not limited to JamVM or HaikuVM. Other platforms for virtualization and emulation of a computer system by PP 101 are possible as well, without departing from the scope of the present disclosure, where firmware 106 can be compiled or formatted to operate on PP 101 operating as a host for the virtualized computer system. In exemplary embodiments, firmware 106 can comprise an
30 application where PP 101 operates as a process virtual machine or an application virtual machine. The environment in which firmware 106 operates can also be referred to as a managed runtime environment (MRE). Firmware 106 could also comprise a primary firmware for the operation of PP 101 as a host computer for the operation of secondary firmware.

Firmware 106 can comprise compiled software or machine executable instructions for either
35 (i) a processor or (ii) a virtual machine in PP 101, and may also be referred to herein as an “image”. In other words, although (A) firmware may traditionally refer to machine executable programming

instructions that provides low-level or hardware control over computing hardware, such as memory and physical interfaces, as contemplated herein, (B) “firmware” can comprise higher level software written for a virtual machine. In addition, the computing environment of a primary platform can require secure functions such as writing and reading cryptographic keys for a firmware 106 specially designated protected memory, and thus firmware 106 comprising high level software may include features similar to traditional firmware. Further, firmware may be traditionally associated with machine executable instructions that are read from a read only memory, and firmware 106 comprising software that is loaded into primary platform 101 can have features after loading in PP 101 that are similar to traditional firmware, such as firmware 106 not being readily modified by an insecure processor in device 102. In any case, although “firmware 106” is described herein as firmware, “firmware 106” can comprise any collection of machine executable instructions which can be loaded and operated by primary platform 101. Similarly, the GSMA PP Requirements document and GP OFL document refer to the collection of machine executable code for a primary platform 101 as “firmware”.

As depicted in Figure 1a, an image delivery server 103 (or “server 103”) can include a public key PK.IDS1.ECKA 103a, a public key PK.IDS2.ECDSA 103b, a record certificate function 103e, an OCSP client 103f, a public key validation function 103g, a key exchange A’ 103h algorithm, a key exchange B’ 103i algorithm, and a OCSP query repeat 103j function. Although not depicted in Figure 1a, a server 103 could record the corresponding private keys for public keys PK.IDS1.ECKA 103a and PK.IDS2.ECDSA 103b. Server 103, PP 101, and the other nodes depicted in Figure 1a can also commonly share cryptographic parameters and cryptographic algorithms as specified in the depicted certificates for Figure 1a and Figure 1c below. The record certificate function 103e for server 103 can record the certificate for both (i) PK.IDS1.ECKA 103a comprising CERT.IDS1.ECKA 103c and (ii) PK.IDS2.ECDSA 103b comprising CERT.IDS2.ECDSA 103d, where the public keys are sent to certificate issuer 104 and the certificates are received from certificate issuer 104.

OCSP client 103f can comprise a client to operating according to the online certificate status protocol (OCSP) according to IETF RFC 6960 titled “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP”, which is hereby incorporated by reference in its entirety. OCSP client 103f could support future or related standards for periodically verifying or determining if a previously issued certificate has been revoked or is invalid. OCSP client 103f can submit a query 109 for a certificate such as CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d to CI 104 and receive a response 110 from CI 104. An exemplary query 109 and response 110 for an OCSP client 103f is also depicted and described in connection with Figure 2a below. Note that response 110 from OCSP responder 104d includes a signed time T2 104e, where signed time T2 104e can comprise both a digital signature portion and a plaintext time value for T2 104e-1 (in

Figure 2a). The value for T2 104e-1 can comprise a duration of validity confirmation of CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d over a shorter period of time than the expiration time of the certificate, such as an exemplary several days or few weeks, instead of the certificate expiration time T1 104c. By receiving the value for T2 104e in an authenticated manner and using the subsequent steps described herein, PP 101 can reasonably be assured that a certificate associated with IDS server 103, such as CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d has not been revoked.

A key exchange A' 103h algorithm for server 103 can use a received ephemeral ECC public key PKE1 from a message 117 and the secret or private key for PK.IDS.ECKA 103a in order to mutually derive a symmetric ciphering key M1 and MAC key H1 with PP 101. A key exchange A' 103h can include the step "KS13=DERIVE(SK.IDS1.ECKA)[PK.OFL.ECDHE]" in section 4.1.2.3 of the GP OFL document on page 36. Server 103 can then derive an ephemeral PKI key pair that includes an ephemeral public key of PKE2 and a corresponding ephemeral private key. Server 103 can conduct a key exchange B' 103i using the corresponding ephemeral private key and the public key from CERT.PN.ECDSA 101a-2. A key exchange B' 103i can include the step "KS26=DERIVE(SK.IDS2.ECDHE)[PK.OFL.ECDHE]" in section 4.1.2.3 of the GP OFL document on page 36. Although not depicted in Figure 1a, a server 103 can also conduct the other steps from section 4.1.2.3 of the GP OFL document on page 36 along with key exchange A' 103h and key exchange B' 103i in order to prepare and process a message 118 with an encrypted firmware 106* in a secure and authenticated manner.

A server 103 can include an OCSF query repeat function 103j. The OCSF query repeat function 103j can determine if a time value T2 104e from an OCSF response 110 is less than a timestamp T3 108b-1 from an image owner 105. If time value T2 104e is less than timestamp T3 108b-1, then server 103 can repeat the OCSF query 109. Time value T2 104e-1 could be less than timestamp T3 108b-1 for embodiments where T2 104e-1 and timestamp T3 108b-1 are close together and clocks for nodes in a system 100 are not fully synchronized, such as an exemplary minute apart in time. Additional details for the operation of an OCSF query repeat function 103j with associated time values will be described below in the overall sequence of message flows between the nodes.

A system 100 can also include a certificate issuer (CI) 104. CI 04 can comprise a collection of servers to issue certificates and also verify certificate status after issuance. CI 104 can include private or secret keys SK.CI.ECDSA 104a and SK.CI-OCSF.ECDSA 104b, a verify and sign function 104c, certificates CERT.IDS1.ECKA 103c and CERT.IDS2.ECDSA 103d, and an OCSF Responder 104d. Secret key SK.CI.ECDSA 104a can comprise a root private key for the root certificate and public key in CERT.CI.ECDSA 101a-3 recorded in primary platform PKI keys 101a as depicted in Figure 1c below. Secret key SK.CI-OCSF.ECDSA 104b can comprise a private key

corresponding to the public key in CERT.CI- OCSP.ECDSA 101a-4 recorded in primary platform PKI keys 101a as depicted in Figure 1c below. The verify and sign function 104c can receive public keys from IDS 103 such as PK.IDS1.ECKA 103a and PK.IDS2.ECDSA 103b and use the secret key SK.CI.ECDSA 104a in order to generate certificates of CERT.IDS1.ECKA 103c and CERT.IDS2.ECDSA 103d, respectively. Note that CERT.IDS2.ECDSA 103d includes an expiration time T1 104c, which represents the time for end of the validity of the certificate.

CI 104 can operate an OCSP responder 104d. OCSP responder 103d can comprise a responder to operating according to the online certificate status protocol (OCSP) according to IETF RFC 6960 titled “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP”, which is hereby incorporated by reference in its entirety. OCSP responder 104d could support future or related standards for periodically confirming or reporting if a previously issued certificate has been revoked or is invalid. OCSP responder 104d can receive a query 109 for a certificate such as CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d from a server such as IDS server 103 and generate a response 110 for the query 109. An exemplary query 109 and response 110 for an OCSP responder 104d is also depicted and described in connection with Figure 2a below.

In exemplary embodiments, CI 104 uses private key SK.CI-OCSP.ECDSA 104b for creating and generating the signed time value T2 104e instead of SK.CI.ECDSA 104a, and in this manner the SK.CI.ECDSA 104a can remain more secured (such as 104a being recorded offline on a different server). Consequently, PP 101 can use CERT.CI- OCSP.ECDSA 101a-4 recorded in PKI keys 101a (shown in Figure 1c) to verify the signed time value T2 104e. In other exemplary embodiments, CI 104 can use the SK.CI.ECDSA 104a to generate a signature for signed time value T2 104e, and in these embodiments TRE 113 and PP 101 can use the CERT.CI.ECDSA 101a-3 (shown in Figure 1c) to validate the signed time value T2 104e using steps equivalent to step 108b in Figure 2b below.

A system 100 can also include an image owner 105. Image owner 105 can comprise a collection of servers to create or record firmware 106 images for PP 101 and also process or issue authentication tokens to enable PP 101 to load and operate firmware 106. The authentication tokens can include information for permission or rights of the TRE 113 to operate with the firmware 106. Note that a TRE manufacturer could also operate as an image owner 105, and other possibilities exist as well for the entity that performs the functionality of an image owner 105. An image owner 105 in a system 100 can be compatible with an image owner as described in the GP OFL document. Although a single firmware 106 image is depicted in Figure 1a, an image owner 105 could record a plurality of firmware 106 images. Image owner 105 can receive signed timestamp T3 108b from an OFL authority 107. Note that a signed timestamp 108b can include both a plaintext time value T3 108b-1 and a digital signature from a timestamp authority (TSA). For some exemplary embodiments, a signed timestamp 108b can include both a plaintext time value T3 108b-1 and a

transaction identity (ID) 117x, where (i) the transaction ID 117x can be generated by TRE 113. Additional details regarding using a signed timestamp T3 108b are provided in Figure 2b below.

A system 100 can also include an OFL authority 107. OFL authority 107 can comprise an entity or collection of servers for granting the rights to an Image Owner for managing an image by using OFL. Although not depicted in Figure 1a, an OFL authority 107 can also send an image owner 105 an authentication token with access rights. In some embodiments, such as when an OFL state becomes “locked”, an image owner 105 can become the OFL Authority 107. As depicted in Figure 1a, and OFL authority can operate with a time stamp authority (TSA) 108. TSA 108 could operate according to the American National Standards Institute standard X9.95 for creating signed timestamps, and other possibilities exist as well. Although depicted as operated by OFL authority 107, TSA 108 could operate outside of OFL authority 107 and be associated with OFL authority 107 or an IDS 103.

TSA 108 can record a secret key SK.TSA 108a corresponding to the public key in CERT.TSA 101a-5 from PP 101 PKI keys 101a in Figure 1c. TSA 108 can operate a clock for tracking date and time that is closely synchronized with established time standards. TSA 108 can create a signed timestamp T3 108b using the SK.TSA 108a. The signed timestamp T3 108b can comprise a digital signature portion and a plaintext timestamp value of T3 108b-1. For some embodiments as depicted in Figure 1a, TSA 108 could include a transaction ID 117x with the signed timestamp T3 108b, where (i) the TSA 108 could receive the transaction ID 117x from a message 117b from IDS 103 and (ii) the IDS 103 could receive the transaction ID 117x from TRE 113 in a message 117. As depicted in Figure 1a, the signed timestamp T3 108b can be received by image owner 105, and image owner 105 can send server 103 a message 112 with both the encrypted firmware 106 and the signed timestamp 108b, along with other data to support an open firmware loader.

The nodes for system 100 depicted in Figure 1a can communicate a series of messages with data in order for PP 101 to receive firmware 106 in a secure and authenticate manner. As described above, IDS server 103 can periodically conduct an OCSP query 109 with CI 104 in order to receive a response 110, where an exemplary response 110 is depicted and described in connection with Figure 2a below. The response 110 can include a digital signature 104e for time value T2 104e-1, where the signed time value T2 104e-1 represents a time where a certificate for IDS 103 such as CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d have not been revoked. In other words, the signed time value T2 104e-1 can comprise a point in time in the future through which the certificate can be considered valid and not revoked, where the point in time is shorter than the certificate expiration time T1 104c. In exemplary embodiments the signed time value T2 104e-1 could be an exemplary week into the future, while certificate expiration time T1 104c could be several years into the future.

As described above, server 103 could receive firmware 106 image and a digital signature 108b for timestamp value T3 108b-1 from image owner 105 in a message 112. The signed timestamp value T3 108b-1 could originate from a TSA 108. Note that IDS 103 could receive message 112 before or after device 102 and IDS server 103 establish a secure session such as TLS 115 (for embodiments where a transaction ID 117x is not included in digital signature 108b for at least T3 108b-1). For embodiments where a transaction ID 117x is included in digital signature 108b, then message 112 could be received by IDS 103 after (i) the receipt of a message 117 from TRE 113 and (ii) sending or transmitting a transaction ID value 117x to TSA 108 in a message 117b.

Although timestamp value T3 108b-1 (originating from a timestamp authority 108) along with signature 108b is depicted in Figure 1a as being received by server 103 from an image owner 105, timestamp value T3 108b-1 along with signature 108b could be received from another entity for a system 100 besides image owner 105. For some embodiments, server 103 could operate as a timestamp authority and generate timestamp value T3 108b-1 along with signature 108b. Device 102 and server 103 can establish a secure session, which could comprise a TLS session 115. For system 100, server 103 and device 102 may establish a secure session 115, which could comprise establishing a secure communications link between the two servers using protocols such as TLS, IPSec, a virtual private network (VPN), a secure shell (SSH), or similar networking, transport, or application layer technologies in order to establish secure communications between server 103 and device 102.

Secure session 115 is depicted in Figure 1a as “TLS 115” but other methods for establishing secure sessions could be used as well, including Datagram Transport Layer Security (DTLS). Secure session 115 can utilize certificates for server 103 and/or device 102 in order to provide mutual authentication and mutual key derivation for a symmetric encryption key in secure session 115. For some embodiments, the use of a certificate for device 102 in secure session 115 could be omitted, and for these embodiments only server 103 would be authenticated. Other possibilities exist as well for establishing a secure session 108 between server 103 and device 102 without departing from the scope of the present disclosure. Server 103 can begin listening for incoming messages from a device 102 using a physical network interface that provides connectivity to the IP network 114 and server 103 can use a specific port number such as TCP port 443 to listen for incoming data for secure session 115 from a device 102. Note that TLS session 115 is established between device 102 and server 103, and a TRE 113 may not establish the secure session directly, but rather send and receive data over a secure session established by device 102.

As depicted in Figure 1a, server 103 can use an OCSP Query Repeat 103j step after the receipt or generation of a signed timestamp T3 108b-1, such as the receipt in a message 112. For embodiments where (i) a clock to generate T3 108b-1 and (ii) the validity time set by a CI 104 and specified in T2 104e-1 are close together, such as a few seconds apart, there could be instances

where time for received T3 108b is after the time value T2 104e-1. Server 103 could detect that T3 108b-1 is after T2 104e-1 in an OCSF Query Repeat 103j step before sending message 118 and consequently “refresh” or repeat the query 109 with responder 101d in order to obtain a response 110 with a signed time value T2 104e-1 that is after the signed timestamp T3 108b-1.

5 Server 103 can then send device 102 a message 116, where message 116 can include a value for code M and the certificate for server 103 comprising CERT.IDS1.ECKA 103c. Code M can comprise a challenge or nonce from server 103 which will subsequently be signed and used by PP 101 in order to authenticate PP 101 with server 103. Note that in exemplary embodiments, CERT.IDS1.ECKA 103c in message 116 can utilize OCSF stapling, which is also known as the TLS
10 Certificate Status Request extension. The extension is described in section 8, “Certificate Status Request” of IETF RFC 6066 titled “Transport Layer Security (TLS) Extensions: Extension Definitions”. In other words, the certificate 103c in message 116 can include a response 110 from OCSF responder 104d, where a response 110 is also depicted and described in connection with Figure 2a below. An extension for certificate 103c verifying CERT.IDS1.ECKA 103c has not been
15 revoked for an interim period could utilize related or future versions of the standards as well.

Note that CERT.IDS1.ECKA 103c in message 116 can include a first signed time value T2 104e-1 where the digital signature comprises 104e. A second time value T2 104e-1 for CERT.IDS2.ECDSA 103d can be included in message 118 below. In exemplary embodiments, message 116 could also include a CERT.CI- OCSF.ECDSA 101a-4 (not shown in Figure 1a but
20 shown in Figure 1c), where CERT.CI- OCSF.ECDSA 101a-4 can be used to verify the first signed time value T2 104e-1 in the form of digital signature 104e. For embodiments where CERT.CI- OCSF.ECDSA 101a-4 is included in a message 116, then PP 101 can use certificate CERT.CI.ECDSA 101a-3 to validate the certificate CERT.CI- OCSF.ECDSA 101a-4.

Device 102 can receive message 116 and forward data in the message to PP 101 through
25 OFL agent 102x. PP 101 can receive the data from message 116 and conduct steps in section 4.1.2.1 and 4.1.2.2 of the GP OFL document, including a key exchange 101b in order to encrypt CERT.PN.ECDSA 101a-2. PP 101 can then respond to OLF agent 102x with a derived ephemeral public key PKE1 and the encrypted CERT.PN.ECDSA 101a-2. Device 102 using a TRE 113 can also generate a transaction ID 117x, where transaction ID 117x can comprise a pseudo-random
30 number or other unique number to uniquely identify a subsequent message 117 and response 118. Device 102 can then send server 103 a message 117, where message 107 includes the derived ephemeral public key PKE1 and the encrypted CERT.PN.ECDSA 101a-2, and also the transaction ID 117x. Note that message 117 can include other data from steps in section 4.1.2.1 and 4.1.2.2 of the GP OFL document.

35 Server 103 can receive message 117 and conduct the series of steps to verify the received ephemeral public key PKE1 and derive a symmetric ciphering key that was used to encrypt

CERT.PN.ECDSA 101a-2. Server 103 can conduct a public key validation step 103g on received ephemeral public key PKE1 in order to ensure the key is valid and on the selected curve in parameters for CERT.IDS1.ECKA 103a. Step 103g by server 103 can comprise conducting the steps for an ECC Full Public-Key Validation Routine in section 5.6.2.3.2 of FIPS publication SP 800-56A (revision 2) for the received device ephemeral public key PKE1. Alternatively, step 103g can comprise server 103 performing the steps ECC Partial Public-Key Validation Routine in section 5.6.2.3.3 of the same FIPS publication. Other example steps within a public key validation step 103g can comprise (i) verifying the public key is not at the “point of infinity”, and (ii) verifying the coordinates of the point for the public key are in the range $[0, p-1]$, where p is the prime defining the finite field. Other possibilities exist as well for evaluating and validating a received public key is cryptographically secure in a public key validation step 103g, without departing from the scope of the present disclosure.

In exemplary embodiments, a public key validation step 103g can include verifying that a point or public key received in a message such as message 117 is not reused, and in this manner a step 103g can be used to reject messages 299 that could comprise a “replay attack”. Server 101 could record in a server database a list of received PP 101 ephemeral public keys PKE1 for an extended period of time, and if any key PKE1 is reused then message 117 could be rejected. The period of time could be suitable for the application used by TRE 113 and server 103, including potentially a period of years. As contemplated in the present disclosure a PP 101 and server 102 can conduct a public key validation step 103g any time a public key or a point on an elliptic curve is received.

Server 103 can conduct the key exchange A' 103h with the PKE1 and secret key for PK.IDS1.ECKA 103a in order to derive the symmetric ciphering key. Server 103 can use the symmetric ciphering key to decrypt CERT.PN.ECDSA 101a-2 and read and process data using the plaintext certificate. Note that transaction ID 117a can be included either (i) within ciphertext that includes CERT.PN.ECDSA 101a-2 or (ii) as plaintext within message 117 and external to the ciphertext. For some exemplary embodiments as depicted in Figure 1a, server 103 can then send the Timestamp Authority (TSA) 108 the transaction ID 117a in a message 117b.

Server 103 can use the steps in section 4.1.2.3 of the GP OFL document in order to process data for a message 118 with the ciphertext firmware 106*. Server 103 can receive the data for a message 118 to send to device 102 from image owner 105 in a message 112. Message 118 can include both (i) a digital signature 108b for timestamp T3 108b-1 and (ii) the timestamp T3 108b-1. A server 103 can conduct the key exchange step B' 103i when conducting the steps for section 4.1.2.3 of the GP OFL document. Server 103 can then send device 102 a message 118, where message 118 includes an ephemeral public key PKE2, (i) the ciphertext firmware 106* (ii) an encrypted CERT.IDS2.ECDSA 103d, and (iii) a timestamp T3 108b-1 along with the digital

signature 108b for the timestamp T3 108b-1. For some exemplary embodiments, both the timestamp T3 108b-1 and the digital signature 108b for the timestamp T3 108b-1 can be included within encrypted data within message 118, such as within a value M4 from the load function 119 depicted in Figure 1c below.

5 Note that in exemplary embodiments, CERT.IDS2.ECDSA 103d in message 118 can utilize OCSP stapling, which is also known as the TLS Certificate Status Request extension as described above. In other words, the certificate 103d in message 118 can include a response 110 from OCSP responder 104d, where a response 110 is also depicted and described in connection with Figure 2a below. An extension for certificate 103d verifying CERT.IDS2.ECDSA 103d has not been revoked
10 for an interim period could utilize related or future versions of the standards as well. Note that CERT.IDS2.ECDSA 103d in message 118 can include a second time value T2 104e. A first time value T2 104e for CERT.IDS1.ECKA 103c could be included in message 116 above. The timestamp T3 108b-1 can be from TSA 108 and received by server 103 in message 112.

In exemplary embodiments, the timestamp T3 108b can be different than the “timestamp”
15 generated by IDS1 in Figure 4.1 of the GP OFL document. The timestamp generated by IDS1 in the GP OFL document (i) is not signed using a digital signature algorithm such as ECDSA and (ii) verified by TRE 113 using a certificate such as Cert.TSA 101a-5 (depicted in Figure 1c below). In other words, the GP OFL document does not suggest a separate timestamp or verification means other than the generation of a timestamp and encryption with a static public key from an image
20 delivery server. Further, there is no teaching or suggestion in the GP OFL document for using the transaction ID to generate the timestamp depicted in Figure 4.1 of the GP OFL document.

Device 102 can receive message 118 and forward PP 101 in TRE 113 data from the message using OFL agent 102x. PP 101 can receive data in message 118. PP 101 can then conduct a load function 119, where details for a load function 119 are also depicted and described in connection
25 with Figure 1c below. In summary, load function 119 for PP 101 can conduct the series of steps for section 4.1.2.4 of the GP OFL document on page 37 with at least the additional steps depicted in Figure 1a. Note that steps in section 4.1.2.4 of the GP OFL document on page 37 include the steps to derive symmetric ciphering keys and convert the encrypted data in message 118 into plaintext. As depicted in Figure 1a, the use of brackets such as “{ }” depicts that the data in a message is
30 encrypted with a symmetric ciphering key and a symmetric ciphering algorithm.

In a step 119a, PP 101 can verify the digital signature for both (i) the first time value T2 104e for CERT.IDS1.ECKA 103c (in OCSP stapling extensions for the certificate) and (ii) the second time value T2 104e for CERT.IDS2.ECDSA 103d (again in OCSP stapling extensions for the certificate) using CERT.CI- OCSP.ECDSA 101a-4 within PKI keys 101a as depicted in Figure
35 1c. Note that the first time value T2 104e-1 can be received by PP 101 along with the digital signature 104e in a message 116 and the second time value T2 104e-1 can be received by PP 101

along with the digital signature 104e in a message 118. The exemplary logic for a step 119a is also depicted for a load function 119 in Figure 1c below, such as the depicted step of “VERIFY(CERT 101a-4) [Signed T2 104e]”. An exemplary signature verification step using a public key, a value, and a signature is also depicted and described in connection with Figure 2b below. As described above, T2 104e can comprise two portions of a signature and the plaintext time value T2 104e-1, as depicted and described in connection with Figure 2a below. Upon conclusion of a step 119a, PP 101 can record the first and second plaintext time values T2 104e-1 for use in a subsequent time comparison step 119c (where the first plaintext time value T2 104e-1 was with CERT.IDS1.ECKA 103c and the second plaintext time value T2 104e-1 was with the CERT.IDS2.ECDSA 103d).

Note that the verification step 119a confirms the signatures 104e for the first time value T2 104e-1 and second time value T2 104e are valid, but PP 101 may need to verify and read a plaintext value for timestamp T3 108b-1 before determining if the first time value T2 104e-1 and second time value T2 104e are acceptable time values (such as determining if time T3 108b-1 is in the future of a time T2 104e, which could be an error or unacceptable state since a certificate may have been revoked after time T2 104e and before time T3 108b). In other words, since PP 101 may not have the functionality to keep local or UTC date and time, additional data and steps can be required in order to determine if CERT.IDS1.ECKA 103c and CERT.IDS2.ECDSA 103d have not been revoked or remain reasonably valid using both (i) first time T2 104e and second time T2 104e, and (ii) time T3 108b.

A step 119b within load function 119 can comprise PP 101 verifying a signature 108b for timestamp T3 108b-1 using a public key from a certificate associated with TSA 108, such as CERT.TSA 101a-5 in PP 101 PKI keys 101a as depicted in Figure 1c below. The exemplary logic for a second step 119a is also depicted for a load function 119 in Figure 1c below, such as the depicted step of “Verify(CERT 101a-5) [Signed T3 108b]”. Step 119b can also comprise PP 101 storing the timestamp T3 108b-1 for use in subsequent steps. Note that the signature 108b for timestamp T3 108b-1 can also be over the transaction ID 117x generated by TRE 113 and PP 101 in the processing of a message 117. Part of a step 119b can include verifying that digital signature for timestamp T3 108b is also over (or includes) the transaction ID 117x. In this manner, TRE 113 and PP 101 can trust that the signature 108b for timestamp T3 108b-1 is current and generated after TRE 113 creates transaction ID 117x and sends message 117. In other words, the use of transaction ID 117x helps prevent a “reply attack” by server 103 simply sending an old timestamp value.

A step 119c within load function 119 can comprise PP 101 comparing the first and second plaintext time values T2 104e-1 and timestamp T3 108b-1. Again, PP 101 may not keep date and time due to limited resources of a TRE 113, and thus can rely on time values with digital signatures that have been previously verified in steps 119a and 119b in order to determine if a certificate remains valid or reasonably determine that a certificate has not been revoked. In step 119c, PP 101

can compare timestamp T3 108b-1 with both the first time T2 104e-1 associated with CERT.IDS1.ECKA 103c (from message 116) and the second time T2 104e-1 associated with CERT.IDS2.ECDSA 103d (from message 118). The time values T2 can represent the “next update” date and time from an OCSP response 110 as depicted in Figure 2a, for both of the certificates 103c and 103d. The status of the certificates in an OCSP response can also be “good” or the equivalent of not being revoked. Time values T2 should be in the future when representing the “Next Update” time within an OCSP response 110 as depicted in Figure 2a (while PP 101 may not track local or global time).

PP 101 can compare the first and second time values T2 104e-1 (from message 116 and message 118) with timestamp T3 108b-1 in a step 119c. If the time values T2 104e-1 are greater than timestamp T3 108b, then T3 108b can be considered in the future and certificates CERT.IDS1.ECKA 103c and CERT.IDS2.ECDSA 103d can be accepted and PP 101 can proceed to step 119d for establishing a firmware session with plaintext firmware 106. Alternatively, PP 101 could determine that timestamp T3 108b is less than either of time values T2 104e-1 in a step 119c in order to accept the certificates 103c and 103d. Although not depicted in Figure 1a, TRE 113 and PP 101 could also use the “timestamp” value from the GP OFL document in Figure 4.1 (“GP timestamp”) to additionally confirm proper time values. For example, the PP 101 could confirm that the GP timestamp is less than the first and second time values T2 104e for certificates 103c and 103d, respectively.

Note that PP 101 should perform the other steps for a load function 119 such as certificate verification, at least one ECDH key exchange to derive a symmetric key, and symmetric decryption of ciphertext in message 118 in order to read an authenticated the plaintext firmware 106. Exemplary additional steps are depicted and described in connection with Figure 1c, and also the exemplary additional steps could comprise the steps within section 4.1.2.4 of the GP OFL document. Note that for some embodiments, the use of two separate time values T2 104e-1 and/or two separate certificates such as CERT.IDS1.ECKA 103c (from message 116) and CERT.IDS2.ECDSA 103d (from message 118) could be omitted, and the use of a single time value T2 104e-1 for a single certificate could be use in a step 119c.

For a first portion of step 119c, if timestamp T3 108b-1 is either in the future or greater than the first time value T2 104e-1 (e.g. from a message 116), then PP 101 cannot be reasonably assured or confirm that CERT.IDS1.ECKA 103c remains valid and the certificate could potentially be revoked. An equivalent determination in a step 119c can be if the first time value T2 104e-1 is less than or at a past time compared to timestamp T3 108b-1. For this case PP 101 can stop the firmware 106 load function 119 and conduct other steps or queries with server 103 in order to notify or attempt to rectify the error state. In other words, PP 101 may not be able to confirm CERT.IDS1.ECKA 103c remains valid at the time of timestamp T3 108b-1. Alternatively PP 101

could determine that the first time value T2 104e-1 is in the future or greater than timestamp T3 108b-1, and thus CERT.IDS1.ECKA 103c was reasonably valid and not revoked at the time timestamp T3 108b-1 was created. If the first time value T2 104e-1 is in the future or greater than timestamp T3 108b-1, then PP 101 can proceed with a second portion of step 119c before loading
5 firmware 106 and step 119d for creating a firmware session to begin operating with the firmware 106 (after the next check of the second time value T2 104e-1 in the next paragraph).

For a second portion of step 119c, if timestamp T3 108b-1 is in the future or greater than or after a second time value T2 104e-1 associated with certificate 103d, then PP 101 cannot be reasonably assured or confirm that CERT.IDS2.ECDSA 103d has been revoked. An equivalent
10 determination in a step 119c can be if the second time value T2 104e-1 is less than or at a past time compared to timestamp T3 108b. For this case PP 101 can stop the firmware 106 load function 119 and conduct other steps or queries with server 103 in order to notify or attempt to rectify the error state. In other words, PP 101 cannot confirm CERT.IDS2.ECDSA 103d remains valid at the time of timestamp T3 108b-1.

Alternatively, for a second portion of step 119c, PP 101 could determine that the second time
15 value T2 104e-1 is in the future or greater or after timestamp T3 108b-1, and thus CERT.IDS2.ECDSA 103d was reasonably valid and not revoked at the time timestamp T3 108b-1 was created. If the second time value T2 104e-1 is in the future or greater than or after timestamp T3 108b-1 (and the first portion of step 119c is also successful), then PP 101 can proceed with
20 loading firmware 106 and step 119d for creating a firmware session to begin operating with the firmware 106. The conclusion of a step 119d for PP 101 can comprise the exemplary state of loaded firmware in state 3 of loaded firmware in Figure 2-5 of section 2.4 "Loading Procedure" of the GP OLF document. Other examples exist as well for (i) loading firmware 106 after a step 119c to compare time values and (ii) establishing a firmware session in a PP 101 without departing from the
25 scope of the present disclosure.

Figure 1b

Figure 1b is a graphical illustration of hardware, firmware, and software components for a device, including a tamper resistant element with a primary platform, in accordance with exemplary
30 embodiments. Figure 1b is illustrated to include many components that can be common within a device 102, and device 102 may also operate in a wireless configuration in order to connect with a wireless network. In a wireless configuration, the physical interface 102a of device 102 may support radio-frequency (RF) communications with networks including a wireless network via standards such as GSM, UMTS, mobile WiMax, CDMA, LTE, LTE Advanced, 5G, and/or other mobile-
35 network technologies. In a wireless configuration, the physical interface 102a may also provide

connectivity to local networks such as 802.11 WLAN, Bluetooth, Zigbee, or an IEEE 802.15.4 network, among other possibilities. In a wireless configuration, device 102 could use a physical interface 102a connected with both a wireless WAN and wireless LAN simultaneously. In a wired configuration, the physical interface 102a can provide connectivity to a wired network such as through an Ethernet connection or USB connection.

The physical interface 102a can include associated hardware to provide connections to components such as radio-frequency (RF) chipsets, an RF amplifier, an antenna, cable connectors, RF filters, etc. Device drivers 102g can communicate with the physical interfaces 102a, providing hardware access to higher-level functions on device 102. Device drivers 102g may also be embedded into hardware or combined with the physical interfaces. Device drivers 102g can include an OFL agent 102x, which can be utilized by a device 102 and operating system 102h in order to read and write data to TRE 113, including communicating with a primary platform 101 within TRE 113. Device 102 may preferably include an operating system 102h to manage device drivers 102g and hardware resources within device 102. The operating systems described herein can also manage other resources such as memory and may support multiple software programs or software libraries operating on device 102, including applications that communicate with PP 101 through a device driver 102g.

The operating system 102h can include Internet protocol stacks such as a User Datagram Protocol (UDP) stack, Transmission Control Protocol (TCP) stack, a domain name system (DNS) stack, etc., and the operating system 102h may include timers and schedulers for managing the access of software to hardware resources, including TRE 113. The operating system shown of 102h can be appropriate for a low-power device with limited memory and CPU resources (compared to a server 103). Example operating systems 102h for a device 102 includes Linux, Android® from Google®, iOS from Apple®, Windows® Mobile, or Open AT® from Sierra Wireless®. Additional example operating systems 102h for device 102 include eCos, uC/OS, LiteOs, Contiki, OpenWRT, Raspbian, and other possibilities exist as well without departing from the scope of the present disclosure.

A device program 102i may be an application programmed in a language such as, but not limited to, C, C++, Java, and/or Python, and could provide functionality to support M2M applications such as remote monitoring of sensors and remote activation of actuators. A device program 102i could also comprise an application for a mobile phone, table, personal computer, or the like. Device program 102i could also be a software routine, subroutine, linked library, or software device, according to one preferred embodiment. As contemplated herein, a device program 102i may be an application operating within a smartphone, such as an iPhone® or Android®-based smartphone, and in this case device 102 could comprise the smartphone. The application functioning as a device program 102i could be downloaded from an “app store” associated with the

smartphone. Device program 102i can include secure session library 102y, which can provide the functionality or “System on a Chip” (SOC) 109 instructions for conducting secure session 115.

SOC 109 can include a plurality of processing cores appropriate for typically low power consumption requirements for a device 102 (compared to a server 103), and may also function as a microcontroller. The processing cores can include at least one processor core for device 102 such as an ARM® based processor or an Intel® based processor such as belonging to the Atom or MIPS family of processors, and other possibilities exist as well including exemplary ARM Cortex-M processors and cores. A single or set of processor or processing cores could operate as a processor for device 102 and a second processor could operate as a TRE 113. Processing cores within SOC 109 can also comprise specialized processing cores, such as including hardware based cryptographic accelerators, field-programmable gate arrays (FPGAs), and other possibilities exist as well without departing from the scope of the present disclosure. For some embodiments, one core could be “split” or divided via time multiplexing such that during a first time period the processing core operates as a generic processor for device 102, and during a second time period the processing core operates as a primary platform 101. The embodiment from the previous sentence could comprise using an ARM “Trustzone” implementation for the TRE 113 and PP 101 for the system 100 and other systems herein.

Many of the logical steps for operation of device 102 can be performed in software and hardware by various combinations of sensor 102f, actuator 102z, physical interface 102a, device driver 102g, operating system 102h, device program 102i, and SOC 109. Note that device 102 may also optionally include user interface 102j which may include one or more devices for receiving inputs and/or one or more devices for conveying outputs. User interfaces are known in the art for devices 102 and could include a few LED lights or LCD display or OLED display, and thus user interfaces are not described in detail here. User interface 102j could comprise a touch screen if device 102 operates as a smartphone or mobile phone. As illustrated in Figure 1b, device 102 can optionally omit a user interface 102j, since no user input may be required for many “Internet of Things” applications, although a user interface 102j could be included with device 102.

Device 102 may be a computing device or wireless device that includes computer components for the purposes of collecting data from a sensor 102f or triggering an action by an actuator 102y. Device 102 may include a central processing unit (CPU) within SOC 109, a random access memory (RAM) 102e, and a system bus 102d that couples various system components including the random access memory 102e to the processing unit 102b. The system bus 102d may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures including a data bus.

Device 102 may include a read-only memory (ROM) 102c which can contain a boot loader program. Although ROM 102c is illustrated as “read-only memory”, ROM 102c could comprise

long-term memory storage chipsets or physical units that are designed primarily for writing once and reading many times, such as Electrically Erasable Programmable Read-Only Memory (EEPROM). As contemplated within the present disclosure, a read-only address could comprise an address within a read only memory such as a ROM 102c memory address or another hardware address for read-only operations accessible via bus 102d. Changing data recorded in a ROM 102c can require a technician have physical access to device 102, such as removing a cover or part of an enclosure, where the technician can subsequently connect equipment to a circuit board in device 102, including replacing ROM 102c. ROM 102c could also comprise a nonvolatile memory, such that data is stored within ROM 102c even if no electrical power is provided to ROM 102c.

Device 102 can include a SOC 109. SOC 109 can include TRE 113, and additional details for the operation of SOC 109 and TRE 113 is provided in other figures herein. Although TRE 113 is depicted in Figure 1b as operating within SOC 109, TRE 113 could be operated within a removable unit such as an SD card, a SIM card, etc. Or TRE 113 could operate within a separate soldered chip connected to bus 102d. An exemplary removable form factor for TRE 113 could comprise a standard SD card, a mini SD card, a micro SD card, a mini UICC, a micro UICC, or a nano UICC, and other possibilities exist as well without departing from the scope of the present disclosure. SOC 109 can include electrical contacts (such as external pads 109a depicted in Figure 1c) which provide electrical connectivity to bus 102d. SOC 109 can include NAND or NOR flash memory in order to record data when device 102 is not powered, and other nonvolatile memory technologies can be used in a storage unit as well without departing from the scope of the present disclosure.

The inclusion of TRE 113 and the operation of TRE 113 with PP 101 in SOC 109 can add functionality for SOC 109 that is not normally included in commercially available SOC in the market as of 2018, such as with the secure receipt of firmware 106 as described herein. TRE 113 within SOC 109 can include a processor, bus, and memory similar (but with less power and on a smaller scale) equivalent to bus 102d, and ROM 102c, NVM 102s. TRE 113 can perform cryptographic functions using either boot firmware or downloaded firmware 106 such as (i) internally deriving a private key with a corresponding public key such as PKE1 (in message 117 in Figure 1a) in a cryptographically secure manner, (ii) recording the private key in a protected memory such that device 102 or external parties cannot feasibly or cost-effectively read the derived private key, and (ii) conducting key exchanges and encryption/decryption such as key exchange 101b from Figure 1a.

Although the exemplary environment described herein employs ROM 102c, RAM 102e, and nonvolatile memory (NVM) 102s, it should be appreciated by those skilled in the art that TRE 113 could also operate within other types of computer readable media which can store data that is accessible by a device 102, such as memory cards, subscriber identity device (SIM) cards, local miniaturized hard disks, and the like, which may also be used in the exemplary operating

environment without departing from the scope of the disclosure. The memory and associated hardware illustrated in Figure 1b provide nonvolatile storage of computer-executable instructions, data structures, program devices, device program 102i, device drivers 102g, and other data for computer or device 102. Note the device 102 may include a physical data connection at the physical interface 102a such as a miniaturized universal serial bus adapter, firewire, optical, or other another port and the computer executable instructions such as device program 102i, operating system 102h, or device driver 102g can be initially loaded into memory such as ROM 102c or NVM 102s through the physical interface 102a before device 102 is given to an end user, shipped by a manufacturer to a distribution channel, or installed by a technician.

Further, device program 102i, operating system 102h, or device driver 102g can be separately loaded into NVM 102s before or after distribution of device 102. In some exemplary embodiments, applications or programs operating within device 102 can be given limited or restricted access to TRE 113 in order to support the applications or programs. For example, a mobile payment application operating a device program 102i could authenticate either device 102 or a user with keys recorded in TRE 113 and a firmware 106. Device program 102i could provide a graphical user interface (GUI) to a user through user interface 101j. Other possibilities exist as well for a device program 102i to operate in conjunction with keys and identities recorded in TRE 113 without departing from the scope of the present disclosure.

A number of program devices may be stored in RAM 102e, ROM 102c, or NVM 102s, including an operating system 102h, device driver 102g, an http client (not shown), a DNS client, and related software. TRE 113 can record program devices as well, where the program devices in TRE 113 may be focused on cryptographic operations and functions conducted within TRE 113 in support of the operation of device 102. A firmware 106 depicted and described in connection with Figure 1a and other figures herein can comprise a program device. Program devices include routines, sub-routines, programs, objects, components, data structures, etc., which perform particular tasks or implement particular abstract data types. Aspects of the present disclosure may be implemented in the form of (i) a device program 102i which are executed by the device 102 working in conjunction with (ii) firmware 106 on TRE 113 and PP 101 to authenticate device 102 with a server using public key infrastructure. In exemplary embodiments, program devices for TRE 113 in SOC 109 can include cryptographic algorithms to operate with the cryptographic parameters specified by certificates recorded in the nodes of system 100, such as with PKI keys 101a.

A user may enter commands and information into device 102 through an optional user interface 102j, such as a keypad, keyboard (possibly miniaturized for a mobile phone form-factor), and a pointing device. Pointing devices may include a trackball, an electronic pen, or a touch screen. A user interface 102j may also include a display (not shown) such as a device screen. A display may also be connected to system bus 102d via an interface. The display can comprise any type of display

devices such as a liquid crystal display (LCD), a plasma display, and an organic light-emitting diode (OLED) display. Device 102 may also include a camera (not shown) connected to or integrated with device 102 through a physical interface 102a, and the camera can comprise a video camera for the device 102 to collect sensor data that includes video or images. The camera (not shown) can be a
5 CCD (charge-coupled device) camera, a CMOS (complementary metal-oxide-semiconductor) camera, or a similar device to collect video or camera input including QR codes. Other arrangements could be used as well, without departing from the disclosure.

The device 102, comprising a computer, may operate in a networked environment using logical connections to one or more remote computers, such as servers. Servers communicating with
10 device 102 can also function as a general purpose server to provide files, programs, disk storage, remote memory, and other resources to device 102 usually through a networked connection. Additional remote computers with which device 102 communicates may include another device 102 or mobile device, an M2M node within a capillary network, a personal computer, other servers, a client, a router, a network PC, a peer device, a wireless network, or other common network nodes.
15 The servers or networks communicating with device 102 or a remote computer typically includes many of the elements and electrical components described above relative to the device 102, including a CPU, memory, and physical interfaces. It will be appreciated that the network connections shown throughout the present disclosure are exemplary and other means of establishing a wireless or wired communications link may be used between mobile devices, computers, servers,
20 corresponding nodes, and similar computers. The operation of a TRE 113 within device 102 with a firmware 106 can be utilized to authenticate a device 102 in each or any of the above described networking environments.

The device program 102i operating within device 102 illustrated in Figure 1b and communicating with TRE 113 can provide computer executable instructions to hardware such as
25 SOC 109 through a system bus 102d in order for a device 102 to (i) transmit and receive data with a service provider, (ii) monitor a sensor and/or change the state of an actuator 102y, (iii) send or receive packets with a server or network, and (iv) authenticate with a server, thus allowing the server to remotely monitor or control device 102 in an authenticated and secure manner. The device program 102i can enable the device 102 to authenticate and communicate with a server by recording
30 data in memory such as RAM 102e, where the data can include sensor data, a destination IP address number, a packet or packet header value, an encryption or ciphering algorithm and key, a digital signature and public key, etc, where cryptographic operations or calculations for the device program 102i can be performed by TRE 113 using firmware 106. The data recorded in RAM 102e can be subsequently read by the operating system 102h or the device driver 102g. The operating system
35 102h or the device driver 102g can write the data to a physical interface 102a using a system bus 102d in order to use a physical interface 102a to send data such as a digital signature for

authentication to a server using the Internet 114. In exemplary embodiments, the digital signature can be generated or processed in the TRE 113 using a PP 101 and firmware 106. Alternatively, the device program 102i can write the data directly to the physical interface 102a using the system bus 102d.

5 In general, digital signatures for authentication with a server can be performed in TRE 113, where the digital signature output is transferred from TRE 113 to RAM 102e using a device driver 102g before being transmitted from device 102 to a server through the IP network 114. The data recorded in RAM 102e such as a digital signature can be subsequently read by the operating system 102h or the device driver 102g. Note that device driver 102g can include OFL agent 102x in order
10 to communicate with TRE 113. Thus, OFL agent 102x can be a device driver 102g specifically for TRE 113. The operating system 102h or the device driver 102g can write the data to a physical interface 102a using a system bus 102d in order to use a physical interface 102a to send data such as a digital signature for authentication to a server using the Internet 114. Alternatively, the device program 102i can write the data directly to the physical interface 102a using the system bus 102d.
15 Other possibilities exist as well without departing from the scope of the present disclosure.

The device program 102i or operating system 102h using TRE 113 and PP 101 with firmware 106 can include steps to process the data recorded in memory such as encrypting data, selecting a destination address, or encoding sensor data acquired by (i) a sensor 102f or (ii) through a physical interface 102a such as a thermocouple, shock or vibration sensor, light sensor, or global
20 positioning system (GPS) receiver, etc. The device 102 can use the physical interface 102a such as a radio to transmit or send (i) the data from a sensor or (ii) a digital signature from TRE 113 to a wireless network. For those skilled in the art, other steps are possible as well for a device program 102i or operating system 102h to collect data from either (i) a sensor 102f or (ii) a TRE 113 and send the data in a packet without departing from the scope of the present disclosure.

25 Conversely, in order for device 102 to receive a packet or response from server, which could include a challenge or nonce in order to authenticate a device 102 with the server, the physical interface 102a can use a radio to receive the challenge or nonce from a wireless network. The challenge or nonce received from the server through the wireless network could comprise a random number or a pseudo random string of digits, bits, and/or characters. The received data can include
30 information from a server and may also comprise a datagram, a source IP address number, a packet or header value, an instruction for device 102, an acknowledgement to a packet that device 102 sent, a digital signature, and/or encrypted data. The operating system 102h or device driver 102g can use a system bus 102d and SOC 109 to record the received data such as a challenge or nonce from a server in memory such as RAM 102e, and the device program 102i or operating system 102h may
35 access the memory in order to process the received data and determine the next step for the device 102 after receiving the data.

Processing the received data from a server to device 102 could include deciphering or decrypting received data by TRE 113 with a key recorded in TRE 113, sending the challenge or nonce to the TRE 113, reading an instruction from a server, or similar transformations of the received data. The steps within the paragraph above may also describe the steps a device program 102i can perform in order to receive a packet. For those skilled in the art, other steps are possible as well for a device program 102i or device 102 to receive a packet or challenge or nonce from a server without departing from the scope of the present disclosure. A server described herein without the designation of “server 103” or IDS 103 can comprise a different server than server 103 communicating with device 102 in support of an application operating as a device program 102i (e.g. a server supporting mobile payments, TRE 113 operating an embedded SIM, TRE 113 authenticating device 102 with a wireless network, etc.)

Moreover, those skilled in the art will appreciate that the present disclosure may be implemented in other computer system configurations, including hand-held devices, netbooks, portable computers, multiprocessor systems, microprocessor based or programmable consumer electronics, network personal computers, minicomputers, mainframe computers, servers, and the like. The disclosure may also be practiced in distributed computing environments, where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program devices may be located in both local and remote memory storage devices. In addition, the terms “mobile node”, “mobile station”, “mobile device”, “M2M device”, “M2M device”, “networked sensor”, or “industrial controller” can be used to refer to device 102 as contemplated herein.

Although not depicted in Figure 1b, a server shown above in Figure 1a such as IDS server 103 and other servers as well can include equivalent internal components as device 102 in order to operate. The server 103 in Figure 1a could include a processor similar to SOC 109, with primary differences for the processor server being increased speed, increased memory cache, an increased number and size of registers, the use of a 64 bits for datapath widths, integer sizes, and memory address widths, as opposed to an exemplary 32 bits for processor cores in SOC 109 or an exemplary 32 or 16 bits for a processor in TRE 113. Other possibilities exist as well for the size of datapath widths for a TRE 113 and processing core in device 102 without departing from the scope of the present disclosure. Similarly, RAM in a server could be a RAM similar to RAM 102e in device 102, except the RAM in a server could have more memory cells such as supporting exemplary values greater than an exemplary 16 gigabytes, while RAM in device 102 could support fewer memory cells such as less than an exemplary 8 gigabytes. Non-volatile memory for storage in a server 103 in Figure 1a could comprise disks, “solid state drives” (SSDs) or “storage area networks” (SAN) for a server. For a physical interface 102a, in exemplary embodiments, a server 103 in Figure 1a could use a physical, wired LAN interface such as a high-speed Ethernet or fiber optic connection.

In exemplary embodiments, a device 102 can include the functional capabilities of (i) collecting sensor data, (ii) changing state of an actuator 102z, (iii) communicating the data associated with a sensor or actuator with a wireless network, and/or receiving a challenge or nonce from a server and sending a digital signature. The device driver 102g, operating system 102i, and/or device program 102i could optionally be combined into an integrated system for providing the device 102 functionality. Other possibilities exist as well for the configuration or combination of components illustrated in Figure 1b without departing from the scope of the present disclosure.

Figure 1c

Figure 1c is a graphical illustration of a primary platform, in accordance with exemplary embodiments. A primary platform 101 can include a set of PKI keys 101a and a load function 119. Other data and structures can be included in a PP 101 operating within a TRE 113 without departing from the scope of the present disclosure. Note that a TRE 113 operating a PP 101, such as the TRE 113 depicted in Figure 1a and Figure 1b can include electrical components similar to those depicted for a device 101 in Figure 1b, such as including memory, a processor, and an internal data bus. PP 101 can operate on the electrical components within TRE 113, and the exemplary set of PKI keys 101a can be recorded in nonvolatile memory or read only memory of TRE 113. The set of PKI keys 101a for PP 101 can include certificates of (i) CERT.IDS1.ECKA 103c, (ii) CERT.PN.ECDSA 101a-2, (iii) CERT.CI.ECDSA 101a-3, (iv) CERT.CI-OCSP.ECDSA 101a-4, (v) CERT.TSA 101a-5, (vi) CERT.OFL.ECDSA 101a-6, a secret key SK.OFL.ECDSA 101a-7 corresponding to the public key for a certificate of TRE 113 by server 103 or OFL authority 107, and a public key for a TRE manufacturer of PK.TREM.ECDSA 101a-8. Other keys and certificates can be recorded in a set of PKI keys 101a for a PP 101 without departing from the scope of the present disclosure. The set of PKI keys 101a can be use with cryptographic algorithms for both key exchange 101b and load function 119. Note that PP 101 and TRE 113 record the data for PKI keys 101a in a secure manner, such that device 102 cannot read or write data to PKI keys 101a without specific permission (such as device 102 passing TRE 113 certificate 103c from a message 116).

The certificates in a set of PKI keys 101a without leading numeral designations in Figure 1c can also comprise the certificates for a primary platform as specified in the GP OFL document. The use of these PKI keys without leading numeral designations is also described in section 4.1.2.4 of the load function of the GP OFL document. The certificates in a set of PKI keys 101a depicted in Figure 1c with numeral designations of “for 119” can comprise certificates used by a PP 101 that are not specified or contemplated by the GP OFL document. Some certificates in PKI keys 101a could be recorded during manufacturing of TRE 113, such as CERT.PN.ECDSA 101a-2 and CERT.CI.ECDSA 101a-3. Other keys in PP 101 could be subsequently received after distribution of

device 102 with TRE 113 to end users, such as CERT.IDS1.ECKA 103c (which could be received via a message 116).

CERT.CI- OCSP.ECDSA 101a-4 in PKI keys 101a can comprise a certificate for PP 101 used to validate signed time value T2 104e in a step 119a, where an exemplary signed time value T2 104e is depicted in Figure 2a below. CERT.CI- OCSP.ECDSA 101a-4 could be stored or recorded in PKI keys 101a after (i) PP 101 receives data from message 116 via device 102 and (ii) PP verifies CERT.CI- OCSP.ECDSA 101a-4 using CERT.CI.ECDSA 101a-3 (where CI 104 signs CERT.CI- OCSP.ECDSA 101a-4 using SK.CI.ECDSA 104a). Or, in other embodiments CERT.CI.ECDSA 101a-3 could be written along with CERT.CI- OCSP.ECDSA 101a-4 to nonvolatile memory of PP 101 during manufacturing of TRE 113.

Note that CI 104 can use the private key SK.CI-OCSP.ECDSA 104 corresponding to the public key in CERT.CI- OCSP.ECDSA 101a-4 in order to create signed time value T2 104e. In exemplary embodiments, (i) CI 104 uses a different private key than SK.CI.ECDSA 104a to sign time value T2 104e (since a server or computer with SK.CI.ECDSA 104a may be offline and more secured), and consequently (ii) PP 101 uses a public key different from the key in CERT.CI.ECDSA 101a-3 in order to verify signed time value T2 104e such as the signature in Figure 2a, and the different public key can comprise the public key in CERT.CI- OCSP.ECDSA 101a-4. However, PP 101 could use CERT.CI.ECDSA 101a-3 to verify or validate CERT.CI- OCSP.ECDSA 101a-4 (if received in message 116), and thus CERT.CI.ECDSA 101a-3 can be used indirectly to verify signed time value T2 104e.

CERT.TSA 101a-5 in PKI keys 101a can comprise a certificate for PP 101 used to validate signed timestamp T3 108b, where the use of CERT.TSA 101a-5 by PP 101 is depicted and described in connection with Figure 2b below. Timestamp Authority 108 can use a private key of SK.TSA 108a, corresponding to the public key in CERT.TSA 101a-5, in order to create the digital signature 108b for timestamp T3 108b-1. For some exemplary embodiments, the use of a separate certificate CERT.TSA 101a-5 to validate a digital signature 108b for timestamp T3 108b-1 could be omitted, and timestamp T3 108b-1 could be included in data which is signed by a private key corresponding to a different certificate stored by PP 101 in PKI keys 101a. As one example, timestamp T3 108b-1 could be included in data signed by server 103, and PP 101 could rely upon a public key from a verified certificate from server 103 such as a CERT.IDS2.ECDSA 103d in order to validate a digital signature 108b for timestamp T3 108b-1. For embodiments where PP 101 uses a CERT.IDS2.ECDSA 103d in order to validate a digital signature 108b for timestamp T3 108b-1, the received CERT.IDS2.ECDSA 103d can include a signed time value T2 104e-1 and PP 101 can verify the signed time value T2 104e-1 using CERT.CI- OCSP.ECDSA 101a-4 and/or CERT.CI.ECDSA 101a-3.

However, for other embodiments, such as those depicted for a load function 119 in Figure 1c, the use of a separate certificate of CERT.TSA 101a-5 for verifying a digital signature 108b for timestamp T3 108b-1 can be preferred, since relying upon a different certificate (e.g. such as CERT.CI- OCSP.ECDSA 101a-4 and/or CERT.CIECDSA 101a-3) can mean essentially that the timestamp T3 108b-1 is “self signed”. In other words, without (X) a separately signed timestamp T3 108b-1 which is verified by a separate CERT.TSA 101a-5, then (Y) PP 101 may not be able to securely rely upon timestamp T3 108b-1 in order to use signed time value T2 104e-1 to verify that a certificate CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d has not been revoked.

For example with embodiments alternative to those depicted for a load function 119 in Figure 1c, if timestamp T3 108b is within signed data verified by PP 101 using a certificate such as CERT.IDS2.ECDSA 103d, then PP 101 cannot securely or reliably use a time value T2 104e-1 associated with certificate CERT.IDS2.ECDSA 103d, because that means the signature for timestamp T3 108b-1 is essentially “self signed”. If the certificate for CERT.IDS2.ECDSA 103d has been revoked and the private key associated with CERT.IDS2.ECDSA 103d has been compromised, then (i) the compromised private key associated with CERT.IDS2.ECDSA 103d could be used to sign timestamp T3 108b-1 with an earlier date before revocation of CERT.IDS2.ECDSA 103d, and (ii) an earlier signed timestamp T3 108b-1 from before revocation of CERT.IDS2.ECDSA 103d could be sent in a message 118 from Figure 1a. Consequently, PP 101 could incorrectly determine that CERT.IDS2.ECDSA 103d is valid, when it could have been revoked. Thus, in exemplary embodiments PP 101 can use a recorded CERT.TSA 101a-5 from a set of PKI keys 101a in order to validate a signature 108b for timestamp T3 108b-1.

A primary platform 101 can include a load function 119. The load function 119 depicted in Figure 1c can comprise the load function 119 from Figure 1a with exemplary additional details for conducting signature or certificate verification, ECDH key exchange to derive symmetric ciphering keys, and also decryption with an integrity check. The steps in a load function 119 without leading numeral designations in Figure 1c can also comprise the steps for a primary platform to perform to process a message 118 such as the steps specified in the GP OFL document. The use of these steps without leading numeral designations is also described in section 4.1.2.4 of the load function of the GP OFL document.

The steps depicted in Figure 1c with numeral designations of a leading “119” can comprise steps of the present disclosure that are not specified or contemplated by the GP OLF document in order to securely load firmware 106 while also allowing a PP 101 to determine if a certificate has been revoked. Again, note that there is no description of certificate revocation or the use of the term “revocation” or “revoked” or like descriptions for invalidating a certificate in the GP OLF document. However, support for revocation of a certificate can be essential in order to maintain the security of a system 100 from Figure 1a. Further, conventional technology for determining if a certificate has

been revoked does not support revocation for firmware load with a primary platform, since the primary platform normally does not have the ability to independently keep date and time, especially when power is removed from PP 101. Subsequent descriptions for a load function 119 will focus on the additional steps not contemplated or described by the GP OLF document. Steps for a load function without leading numeral designations can be found within the GP OFL document.

A step 119a for a load function 119 in exemplary embodiments can comprise the depicted step of “VERIFY(CERT 101a-4) [Signed T2 104e]”. PP 101 can use the certificate CERT.CI-OCSP.ECDSA 101a-4 from PP keys 101a in order to verify the signed time value T2 104e-1. The signed time value T2 104e-1 can comprise an interim time value before the expiration time T1 104c where CI 104 confirms that a certificate such as CERT.CI.ECDSA 101a-3 remains valid. In exemplary embodiments, the signed time value T2 104e can comprise a digital signature over a time value T2 104e-1, and the time value T2 104e-1 can comprise the plaintext time value for an end date/time of validity for an OSCP stapling, as depicted and described in connection with Figure 2a.

The signed time value T2 104e-1 can be included as the end of an interim time window for certificate validity such as for the signed time value T2 104e-1 depicted in Figure 2a below. The signed time value T2 104e-1 could be included in an OSCP response 110 value that is “stapled” or included with a certificate such as CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d which are received by PP 101 via messages 116 and 118, respectively. Consequently, although a message 118 in Figure 1a depicts a message 118 including a digital signature 104e for time value T2 104e-1, the message 118 can also include the time value T2 104e-1. Conducting an exemplary signature verification step is depicted in Figure 2b below for a different signature using a different key, but the same logic and sequence of steps can be used in a step 119a for PP 101 to verify the signed time value T2 104e. Although not depicted in Figure 1c, for embodiment where PP 101 received CERT.CI-OCSP.ECDSA 101a-4 via device 102 from a message 116, PP 101 could also verify CERT.CI-OCSP.ECDSA 101a-4 using CERT.CI.ECDSA 101a-3 before using CERT.CI-OCSP.ECDSA 101a-4 in a step 119a.

As depicted in Figure 1c, upon conclusion of a step 119a, PP 101 can securely read and store the plaintext time value T2 104e-1 (e.g. the time value T2 separate or without a signature). Note that although a single instance of step 119a is depicted for a load function 119 in Figure 1c, a PP 101 could conduct a step 119a for each certificate received, such as, but not limited to CERT.IDS1.ECKA 103c and CERT.IDS2.ECDSA 103d. Consequently, after a step 119c, PP 101 can store and operate with more than one time value T2 104e-1, and subsequently use each of the time values T2 104e-1 with a subsequent step 119c. For the case where the signature verification fails in a step 119a, then PP 101 could abort the load firmware function 119 and report an error condition.

A step 119b for a load function 119 in exemplary embodiments can comprise the depicted step of “Verify(CERT 101a-5) [Signed T3 108b]”. PP 101 can use the certificate CERT.TSA 101a-5 from PP keys 101a in order to verify the digital signature 108b for timestamp T3 108b-1. As contemplated herein, a “time value” and a timestamp can be considered equivalent. The digital signature 108b for a timestamp T3 108b-1 can comprise a timestamp of recent time from a TSA 108 associated with the time for which data in message 118 is processed by server 103. Although not depicted in Figure 1c for a step 119b, a step 119b can comprise PP 101 verifying a digital signature over both the signed time value T3 108b-1 and the transaction ID 117x.

Since PP 101 or TRE 113 may not reliably internally track date and time, the digital signature 108b for a timestamp T3 108b-1 can represent the time PP 101 should use for checking the validity or interim validity of certificates and other time windows. The inclusion of transaction ID 117x within a digital signature 108b for timestamp T3 108b-1 can be useful for PP 101 to verify that the digital signature 108b over at least T3 108b-1 was generated after the PP 101 sends a message 117 with the transaction ID 117x. Upon conclusion of a step 119b, PP 101 can securely read and store the plaintext time value T3 108b-1 (e.g. the timestamp T3 separate or without a signature). As discussed above regarding the use of a CERT.TSA 101a-5, an independent certificate from server 103 can be used to verify the signed timestamp 108b for preferred embodiments. Although not depicted in Figure 1a, the date/time value for the digital signature 108b of timestamp T3 108b-1 can be sent along with a message 118. For other embodiments, a different recorded certificate for PP 101 in PKI keys 101a, such as a PK.IMO.ECDSA (not depicted in Figure 1c, but included in a primary platform in the GP OFL document) could be used to verify digital signature 108b for timestamp T3 108b-1 (assuming the secret key for PK.IMO.ECDSA was used to generate the digital signature 108b for timestamp T3 108b-1. In general, a certificate used to verify a signed timestamp 108b can be from a different entity than server 103, such as from an image owner 105, an OFL authority 107, or a TRE manufacturer.

A step 119c for a load function 119 can comprise PP 101 comparing the plaintext, verified time value T3 108b-1 with each of the time values T2 104e-1. As depicted in Figure 1c, the time value T3 108b-1 should be less than each time value T2 104e-1. Although only a signed time value T2 104e-1 for CERT.IDS2.ECDSA is depicted in Figure 1c, a PP 101 could also use the verified time value T3 108b-1 to verify a different, previous signed time value T2 104e-1 for CERT.IDS1.ECKA 103c from a message 116 in a load function 119. In other words, the timestamp T3 from data in message 118 and associated with received firmware 106 should be before each time value T2 104e-1. For a step 119c, if PP 101 determines that any time value T2 104e-1 is before timestamp T3 108b-1, then PP 101 cannot confirm that a certificate associated with the time value T2 104e-1 remains valid and has not been revoked. If PP 101 determines that any time value T2 104e-1 is before timestamp T3 108b-1, then PP 101 can abort the load 119 function and report the

error condition. Or, PP 101 could respond with an error code and wait for an updated, valid certificate with an updated signed time value T2 104e-1 before completing the load function 119 and begin operating with the firmware 106, such as establishing a firmware session.

After step 119c then PP 101 can conduct a step 119d and proceed with loading firmware 106 and create a firmware session to begin operating with the firmware 106. The conclusion of a step 119d for PP 101 can comprise the exemplary state of loaded firmware in state 3 of loaded firmware in Figure 2-5 of section 2.4 “Loading Procedure” of the GP OLF document. Other examples exist as well for (i) loading firmware 106 after a step 119c to compare time values and (ii) establishing a firmware session in a PP 101 using firmware 106 from a message 118 without departing from the scope of the present disclosure.

Figure 2a

Figure 2a is a simplified response from a certificate authority to verify certificate validity, in accordance with exemplary embodiments. Although a certificate normally has an ultimate expiration time, nodes in a system 100 can need the ability to check that a certificate has not been revoked. As mentioned above, a primary platform recording and using a plurality of different certificates may record and use certificates with long time frames such as several years into the future. For a system 100 potentially hundreds of servers 103 distributed globally and also connected to a public Internet 114, there is a reasonable chance that some certificates for some servers 103 may need to be revoked before the long expiration time of certificates recorded by a PP 101 in a TRE 113. The use of a protocol such as OCSP and an exemplary response 110 depicted in Figure 2a can support a PP 101 determining if a certificate has been revoked.

As depicted and described in connection with Figure 1a, a server 103 could operate a client for requesting a signed, updated status for a certificate. The client could comprise an OCSP client and server 103 could send the request 109 from Figure 1a in order for CI 104 to generate the exemplary response 110 depicted in Figure 2a. CI 104 could operate an OCSP responder 104d from Figure 1a in order to process the request 109 and generate the response 110. The exemplary data depicted for a response 110 is shown using ECDSA algorithms with an SHA-256 hash algorithm, but other algorithms could be utilized as well.

A response 110 in Figure 2a can include the query 109, such that an entity receiving the response 110 can determine the query 109 used to generate the response. As shown, the query could include hash values over the certificate issuer data and a serial number associated with the certificate being checked, such as a serial number for CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d. The format and logic and algorithms used to create the exemplary data in query 109 within response 110 can be compatible with IETF RFC 6960 and future, related standards, although other, similar

techniques could be used as well to generate an interim confirmation by a certificate issuer that a certificate has not been revoked.

The response data in a response 110 include the certificate issuer information and the certificate serial number, as well as a signed time value T2 104e, which can comprise a digital signature 104e. The plaintext time value T2 104e-1 can be read separately from response 110, although digital signature 104e can be over plaintext time value T2 104e-1. In exemplary embodiments, time value T2 104e-1 can comprise the value of “Next Update” or a time into the future (at the time of CI 104 creating signature 104e) through which a certificate can be considered valid. Further, the status of the certificate can be “good” or “OK” or the equivalent. The response 110 for a certificate such as CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d can be appended to the CERT.IDS1.ECKA 103c or CERT.IDS2.ECDSA 103d for delivery by server 103 to device 102 and PP 101. In other words, (i) the message 116 with a CERT.IDS1.ECKA 103c can include a first response 110 with a first time value T2 104e-1 and first digital signature 104e and (ii) the message 118 with a CERT.IDS2.ECDSA 103d can include a second response 110 with a second time value T2 104e-1 and a second digital signature 104e. After verifying the signed time value T2 104e, as well as any intermediate certificate associated with the signed time value T2 104e, such as CERT.CI- OSCP.ECDSA 101a-4, PP 101 can use plaintext time value T2 104e-1 to compare with verified timestamp T3 108b to determine if the status in response 110 (e.g. “good”) is at the time of timestamp T3 108b. In other words, PP 101 can trust or rely upon the status of a certificate if timestamp T3 108b before the time value T2 104e-1.

Figure 2b

Figure 2b is a flow chart illustrating exemplary steps for verifying a digital signature using PKI keys, parameters, and data input, in accordance with exemplary embodiments. The processes and operations, described below with respect to all of the logic flow diagrams and flow charts may include the manipulation of signals by a processor and the maintenance of these signals within data structures resident in one or more memory storage devices. For the purposes of this discussion, a process can be generally conceived to be a sequence of computer-executed steps leading to a desired result.

These steps usually require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is convention for those skilled in the art to refer to representations of these signals as bits, bytes, words, information, elements, symbols, characters, numbers, points, data, entries, objects, images, files, or the like. It should be kept in mind, however, that these and similar terms are associated with appropriate

physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as listing, creating, adding, calculating, comparing, moving, receiving, determining, configuring, identifying, populating, loading, performing, executing, storing etc. that are often associated with manual operations performed by a human operator. The operations described herein can be machine operations performed in conjunction with various input provided by a human operator or user that interacts with the device, wherein one function of the device can be a computer.

In addition, it should be understood that the programs, processes, methods, etc. described herein are not related or limited to any particular computer or apparatus. Rather, various types of general purpose machines may be used with the following process in accordance with the teachings described herein.

The present disclosure may comprise a computer program or hardware or a combination thereof which embodies the functions described herein and illustrated in the appended flow charts. However, it should be apparent that there could be many different ways of implementing the disclosure in computer programming or hardware design, and the disclosure should not be construed as limited to any one set of computer program instructions.

Further, a skilled programmer would be able to write such a computer program or identify the appropriate hardware circuits to implement the disclosed disclosure without difficulty based on the flow charts and associated description in the application text, for example. Therefore, disclosure of a particular set of program code instructions or detailed hardware devices is not considered necessary for an adequate understanding of how to make and use the disclosure. The inventive functionality of the claimed computer implemented processes will be explained in more detail in the following description in conjunction with the remaining Figures illustrating other process flows.

Further, certain steps in the processes or process flow described in all of the logic flow diagrams below must naturally precede others for the present disclosure to function as described. However, the present disclosure is not limited to the order of the steps described if such order or sequence does not alter the functionality of the present disclosure. That is, it is recognized that some steps may be performed before, after, or in parallel other steps without departing from the scope and spirit of the present disclosure.

The processes, operations, and steps performed by the hardware and software described in this document usually include the manipulation of signals by a CPU or remote server and the maintenance of these signals within data structures resident in one or more of the local or remote memory storage devices. Such data structures impose a physical organization upon the collection of data stored within a memory storage device and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer

programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

A step 119a, and related steps for verifying a digital signature in the present disclosure, can use an ECC based elliptic curve digital signature algorithm (ECDSA) algorithm, and other possibilities exist as well without departing from the scope of the present disclosure. PP 101 can use a signature verification in step 119a in order to validate a signature for a received timestamp 108b. When using ECDSA algorithm in non-deterministic mode for a signature creation, a value of “k” or “r”, which could comprise a random number that can be included with the digital signature 108b. When using a DSA or ECDSA in deterministic mode, such as specified in IETF RFC 6979 and titled “Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)”, which are hereby incorporated by reference, then the requirement for a separately transmitted random number with a digital signature such as signed timestamp 108b (such including as value “k” or “r”) can be optionally omitted, such that “r” can be deterministically calculated based on the message to sign.

In exemplary embodiments, server 103 and other servers in Figure 1a can utilize deterministic ECDSA without also sending a random number along with a digital signature, although the value “r” from the deterministic mode could be sent with the digital signature such as signed timestamp 108b. In other words, a value can be sent with the digital signature such as signed timestamp T3 108b that is deterministically derived and associated with the message to sign. In other exemplary embodiments, a random number can be generated as a derived value for the random number such as “r” sent with a digital signature such as signed timestamp T3 108b.

For a signature verification step 119b, the exemplary message to verify comprises the plaintext timestamp value T3 108b-1, which can be decrypted by PP 101 using a load function 119. Although a load function 119 in Figure 1c above depicts the timestamp value T3 108b-1 as decrypted from a message M4, for some embodiments the timestamp value T3 108b-1 could be sent as plaintext in a message 118. In other words, the timestamp value T3 108b-1 does not need to be encrypted using an encrypted message M4 since a digital signature 108b for the timestamp value T3 108b-1 can separately be included with a message 118 as depicted in Figure 1a.

The message to verify values can be transmitted to PP 101, such as shown for timestamp T3 108b-1 and the digital signature 108b for timestamp T3. Note that in some exemplary embodiments for a step 119b, the message to verify can include both the timestamp T3 108b-1 and the transaction ID 117x, where PP 101 generates the transaction ID 117x for creation of the message 117 as depicted and described in connection with Figure 1a. The message to verify values can be input into a message digest algorithm 230, which could comprise a standard algorithm such as SHA-256, SHA-3, or similar algorithms. The output of message digest algorithm 230 can be input along with parameters and a public key from a certificate such as CERT.TSA 101a-5 into signature verification

algorithm 228. Parameters can specify encoding rules, padding, key lengths, selected algorithms, curve names, and other values or fields necessary to utilize a signature algorithm 228. Parameters can be specified in the certificate. Both a signature creation step and a signature verification step use the same or equivalent values for parameters. Private key SK.TSA 108a can comprise the private
5 key used to create the digital signature 108b for at least the timestamp T3 108b-1. Although not depicted in Figure 2b, a random number for values such as “k” and “r” for ECDSA and related signatures could be input as well into a signature algorithm 228.

A signature algorithm 228 for a step 119b can also use input of the received digital signature 108b for timestamp 108b-1, which can be received from a message 118. The calculation of a
10 digital signature 108b using the public key from a certificate such as CERT.TSA 105a-1 can be compared with the received, digital signature 108b. If the two values for the calculated version of the digital signature 108b and the received version of the digital signature 108b (from message 118) match or are equal, then the received, signed timestamp T3 108b-1 can be considered valid and the plaintext timestamp value of T3 108b-1 can be used with a comparison step 119c and T2 104e-1. If
15 the two values for the calculated version of the digital signature 108b and the received version of the digital signature 108b (from message 118) do not match or are not equal, then the received, signed timestamp T3 108b-1 can be considered invalid and subsequently not used. PP 101 can respond with an error code or status to request a new, different signed timestamp T3 108b-1 and stop the firmware load function 119.

Conclusion

Various exemplary embodiments have been described above. Those skilled in the art will understand, however, that changes and modifications may be made to those examples without departing from the scope of the claims.

CLAIMS

1. A method for a tamper resistant element (TRE) to securely receive firmware, the method comprising the TRE:

5 recording a certificate issuer (CI) certificate, a TRE certificate, and a timestamp authority certificate;

receiving from an open firmware loader (OFL) agent, a first server certificate for a server, wherein the first server certificate includes a signed first time value, and wherein the TRE uses at least the CI certificate to verify the signed first time value;

10 deriving an ephemeral PKI key pair comprising an ephemeral public key and an ephemeral private key;

conducting an elliptic curve Diffie Hellman key exchange with the ephemeral private key and a server public key from the first server certificate in order to derive a symmetric ciphering key;

15 encrypting the TRE certificate using at least the symmetric ciphering key;

sending to the OFL agent the encrypted TRE certificate;

receiving from the OFL agent a message with the firmware, a signed timestamp, and a second server certificate for the server, wherein the second server certificate includes a signed second time value, and wherein the TRE uses at least the CI certificate to verify the signed second time value;

20 verifying the signed timestamp using at least the timestamp authority certificate;

comparing (i) the signed first time value with the signed timestamp in order to confirm validity of the first server certificate and (ii) the signed second time value with the signed timestamp in order to confirm validity of the second server certificate; and

25 loading the firmware for the TRE.

2. The method of claim 1, wherein the first server certificate includes a first online certificate status protocol (OCSP) signature for the signed first time value, and wherein the second server certificate includes a second OCSP signature for the signed second time value.

3. The method of claim 2, wherein the signed first time value comprises a first next update value for an OCSP protocol, and wherein the signed second time value comprises a second next update value for the OCSP protocol.

4. The method of any one of claims 1 to 3, further comprising receiving from the OFL agent the message with a certificate for an OCSP server, wherein the TRE uses the certificate for the OCSP server to verify the signed second time value, and wherein the TRE uses the CI certificate to verify the certificate for the OCSP server.

5. The method of any one of claims 1 to 4, wherein a system on a chip includes the TRE, and wherein the TRE operates a primary platform for loading the firmware.

6. The method of any one of claims 1 to 5, further comprising sending to the OFL agent the encrypted TRE certificate and a transaction identity, and verifying the signed timestamp using at least the timestamp authority certificate and the transaction identity.

7. The method of any one of claims 1 to 6, further comprising, sending, by the OFL agent, the encrypted TRE certificate to an image delivery server (IDS), and receiving, by the OFL agent, the message from the IDS.

8. The method of claim 7, further comprising, receiving, by the IDS (i) the signed timestamp from a timestamp authority, and (ii) the firmware from an image owner.

9. A method for a tamper resistant element (TRE) to securely receive firmware, the method comprising the TRE:

recording a certificate issuer (CI) certificate, a TRE certificate, and a timestamp authority certificate;

receiving from an open firmware loader (OFL) agent, a first server certificate for a server, wherein the first server certificate includes a first online certificate status protocol (OCSP) signature for a first time value, and wherein the TRE uses at least the CI certificate to verify the first OCSP signature;

5 generating a transaction identity (ID);

 sending to the OFL agent at least the TRE certificate and the transaction ID;

 receiving from the OFL agent a message with the firmware, a digital signature, and a second server certificate for the server, wherein the second server certificate includes a second OCSP signature for a second time value, wherein the TRE uses at least the CI certificate to
10 verify the second OCSP signature, and wherein the digital signature is generated over at least the transaction ID and a third time value;

 verifying the digital signature using at least the timestamp authority certificate and the transaction ID;

 comparing (i) the first time value with the third time value order to confirm validity of
15 the first server certificate and (ii) the second time value with the third time value in order to confirm validity of the second server certificate; and

 loading the firmware for the TRE.

10. The method of claim 9, further comprising receiving from the OFL agent the message with the third time value.

20 11. The method of claim 9 or 10, wherein the first time value comprises a first next update value for an OCSP protocol, and wherein the second time value comprises a second next update value for the OCSP protocol.

 12. The method of any one of claims 9 to 11, further comprising receiving from the OFL agent the message with a certificate for an OCSP server, wherein the TRE uses the
25 certificate for the OCSP server to verify the second OCSP signature, and wherein the TRE uses the CI certificate to verify the certificate for the OCSP server.

13. The method of any one of claims 9 to 12, wherein a system on a chip includes the TRE, and wherein the TRE operates a primary platform for loading the firmware.

14. The method of any one of claims 9 to 13, further comprising, sending, by the OFL agent, the TRE certificate to an image delivery server (IDS), and receiving, by the OFL agent, the message from the IDS.

15. The method of claim 14, further comprising, receiving, by the IDS (i) the digital signature from a timestamp authority, and (ii) the firmware from an image owner.

16. A method for a server to securely transmit firmware to a device, the method comprising the server:

recording (i) a first server private key for a first server certificate and (ii) a second server certificate;

using an online certificate status protocol (OCSP) client to receive a first signed timestamp for the first server certificate and a second signed timestamp for the second server certificate;

sending, to the device, the first server certificate and at least the first signed timestamp; receiving a message from the device, wherein the message comprises an ephemeral public key and a first ciphertext with a transaction identity (ID);

conducting a first elliptic curve Diffie Hellman key exchange with the ephemeral public key and the first server private key in order to derive a first symmetric ciphering key;

decrypting the first ciphertext using at least the first symmetric ciphering key;

sending the transaction ID to a timestamp authority;

receiving, from the timestamp authority, a third signed timestamp and a first digital signature, wherein the first digital signature is over at least the transaction ID and the third signed timestamp, and wherein the third signed timestamp is less than both the first signed timestamp and the second signed timestamp;

deriving a second server private key and a second server public key;

conducting a second elliptic curve Diffie Hellman key exchange with the ephemeral public key and the second server private key in order to derive a second symmetric ciphering key;

encrypting the firmware with the second symmetric ciphering key;

5 generating a second digital signature for the second server public key, wherein the second digital signature is verified using the second server certificate; and

sending, to the device, the encrypted firmware, the second server public key, the second server certificate, the second signed timestamp, the third signed timestamp, the first digital signature, and the second digital signature.

10 17. The method of claim 16, wherein the first signed timestamp comprises a first next update value for an OCSP protocol, and wherein the second signed timestamp comprises a second next update value for the OCSP protocol.

15 18. The method of claim 16 or 17, further comprising, sending, to the device, the first server certificate with a certificate for an OCSP server, wherein the certificate for the OCSP server is verified using a certificate issuer (CI) certificate.

19. The method of any one of claims 16 to 18, wherein the first server certificate and the second server certificate are verified using a certificate issuer (CI) certificate.

20. The method of any one of claims 16 to 19, wherein the server comprises an image delivery server (IDS), and wherein the IDS receives the firmware from an image owner.

Figure 1a

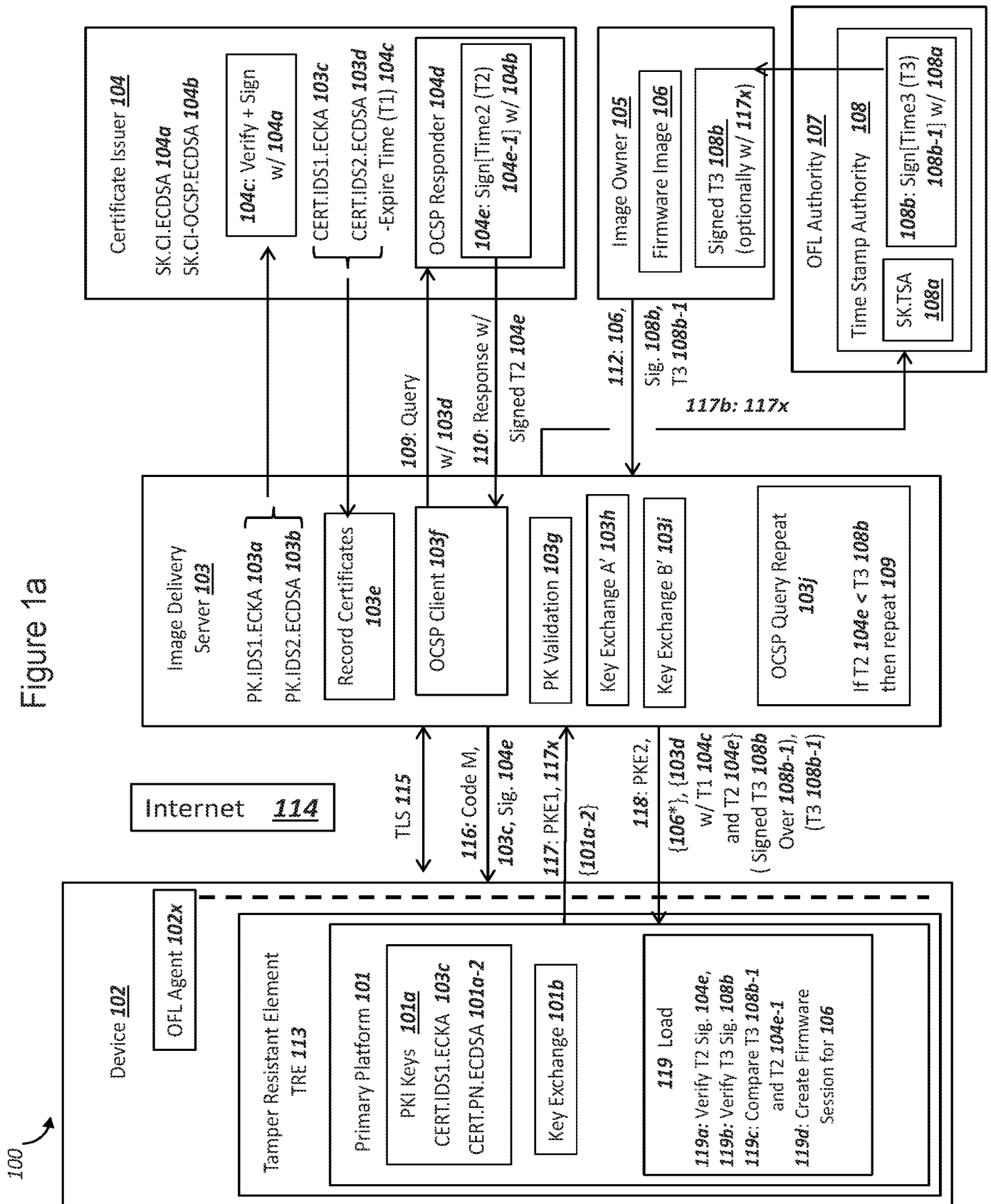


Figure 1c

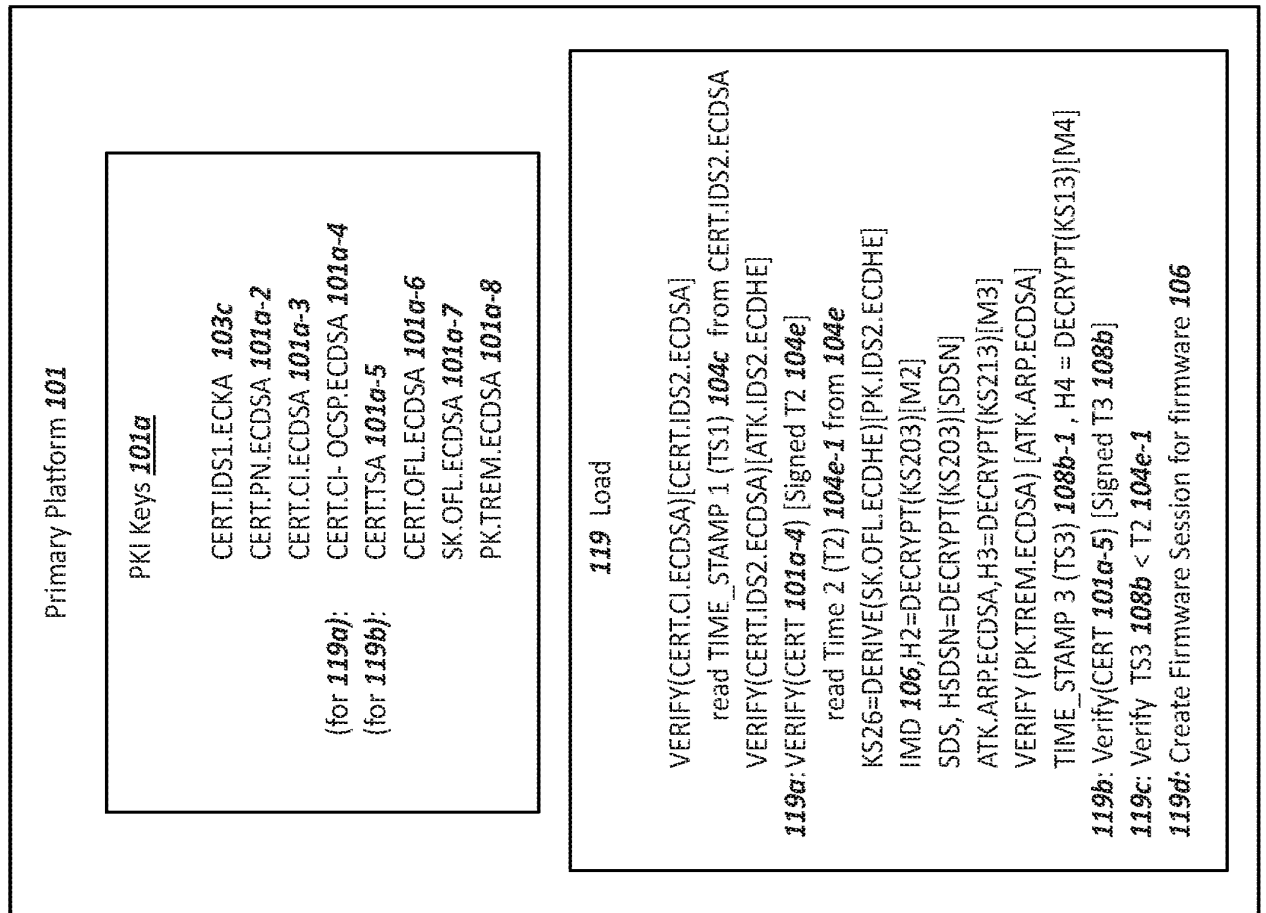


Figure 1b

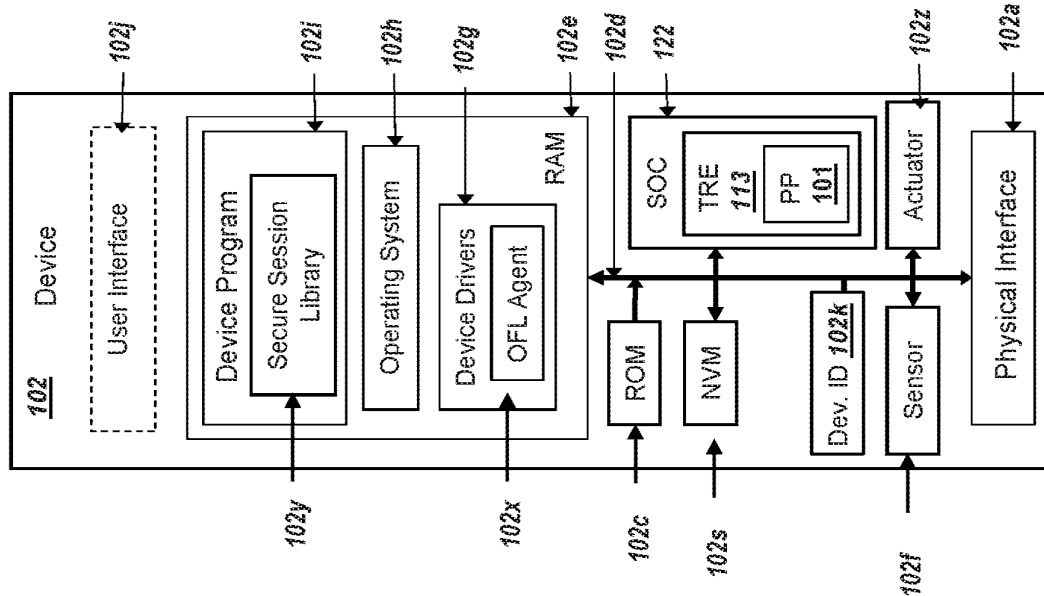


Figure 2a**OCSP Response 110**

Verifying CERT.IDS2.ECDSA with SHA 256 Algorithm ("... " is long text removed)

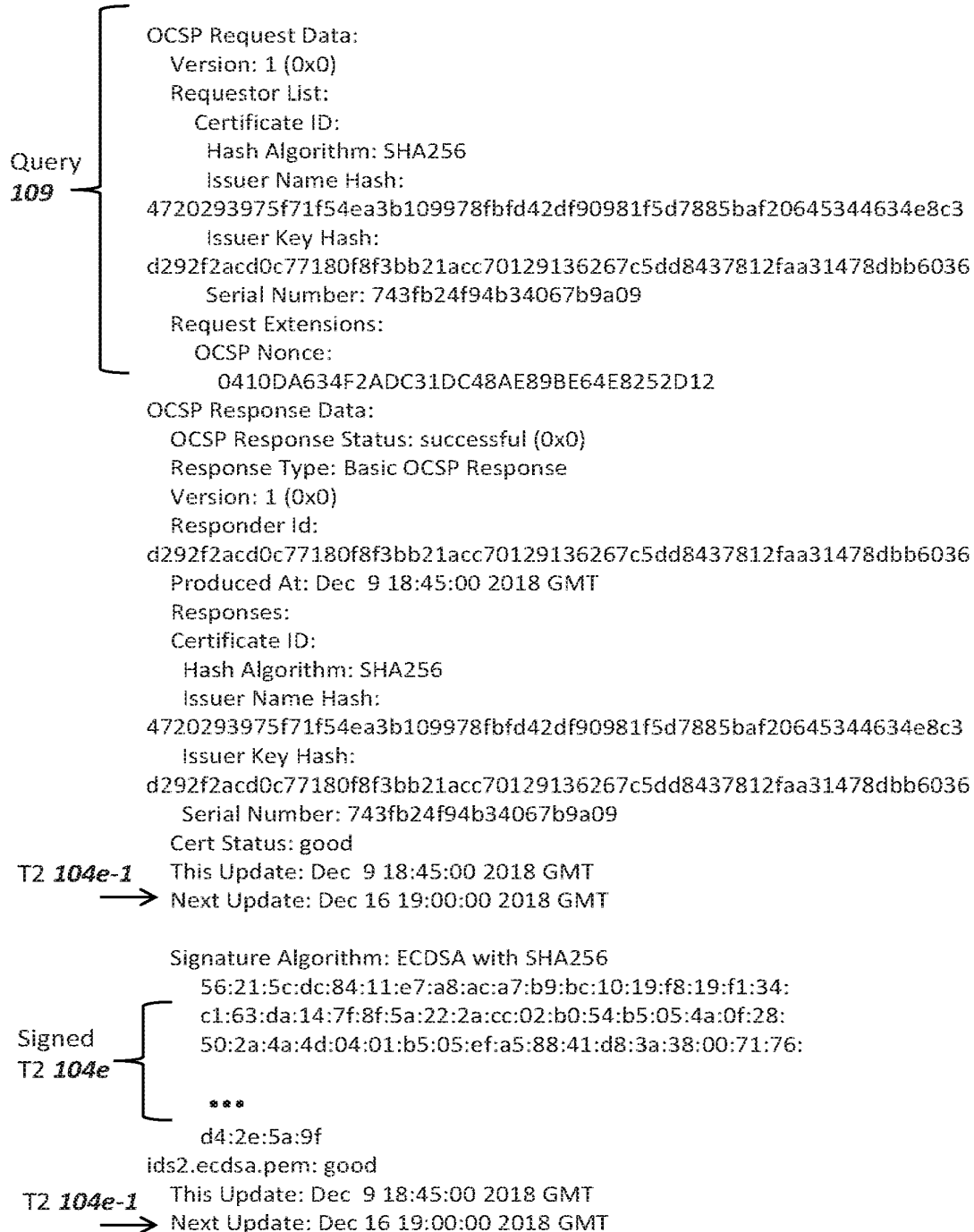
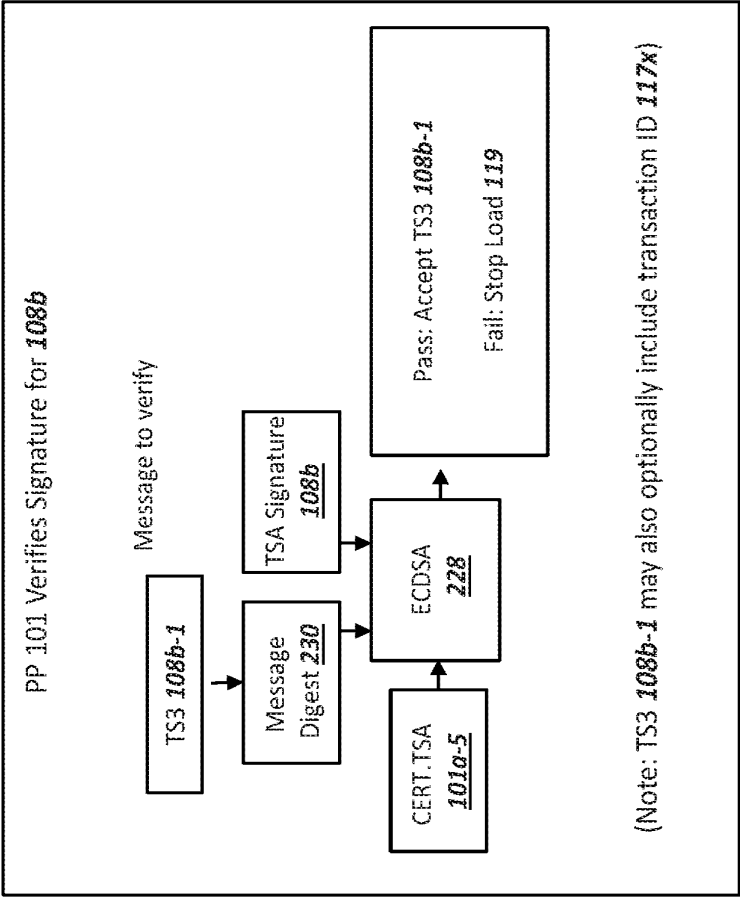


Figure 2b

119b – Time Stamp 3 (TS3) 108b Verification



INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2019/066257

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/64 (2013.01)**H04L 9/14 (2006.01)**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

PatSearch (RUPTO internal), USPTO, PAJ, K-PION, Esp@cenet, Information Retrieval System of FIPS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6009177 A (CERTO LLC) 28.12.1999, col.1, lines 58-60, col.2, lines 4-7, 46-55, col.3, lines 23-36, col.4, lines 59-62, col.11, lines 18-42, col.13, lines 1-6, col.14, lines 29-45, col.21, lines 47-54, col.23, lines 13-30, col.24, lines 1-3, col.27, lines 29-32, col.37, lines 30-35, 60-64, col.38, lines 22-36, col.39, lines 2-6, 24-32, col.42, lines 11-13, col.44, lines 41-44, col.48, lines 51-62	1-4, 9-11, 16-18
A	US 2016/0065370 A1 (ERIC LE SAINT et al) 03.03.2016	1-4, 9-11, 16-18
A	US 7177839 B1 (CERTCO INC) 13.02.2007	1-4, 9-11, 16-18
A	US 2006/0080536 A1 (TIME CERTAIN LLC) 13.04.2006	1-4, 9-11, 16-18



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier document but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

24 March 2020 (24.03.2020)

Date of mailing of the international search report

09 April 2020 (09.04.2020)

Name and mailing address of the ISA/RU:

Federal Institute of Industrial Property,
Berezhkovskaya nab., 30-1, Moscow, G-59,
GSP-3, Russia, 125993
Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37

Authorized officer

A. Kulikov

Telephone No. (495) 531-65-15

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2019/066257

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☒ Claims Nos.: 5-8, 12-15, 19, 20
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

(54) Title of the Invention: **Secure communication with a mobile device**

(51) INT CL: **H04W 4/70** (2018.01) **H04W 8/00** (2009.01)

(21) Application No: **1415927.1**

(22) Date of Filing: **09.09.2014**

(30) Priority Data:

(31) 1316370	(32) 13.09.2013	(33) GB
(31) 1318339	(32) 16.10.2013	(33) GB
(31) 1409652	(32) 30.05.2014	(33) GB
(31) 1409641	(32) 30.05.2014	(33) GB
(31) 1409643	(32) 30.05.2014	(33) GB
(31) 1409663	(32) 30.05.2014	(33) GB
(31) 1415003	(32) 22.08.2014	(33) GB
(31) 1414999	(32) 22.08.2014	(33) GB
(31) 1414997	(32) 22.08.2014	(33) GB

(43) Date of A Publication **08.04.2015**

(72) Inventor(s):

Nick Bone

(73) Proprietor(s):

Vodafone IP Licensing Limited
Babbage House, The Connection, NEWBURY,
Berkshire, RG14 2FN, United Kingdom

(74) Agent and/or Address for Service:

Boult Wade Tennant LLP
Salisbury Square House, 8 Salisbury Square,
LONDON, EC4Y 8AP, United Kingdom

(56) Documents Cited:

WO 2014/071885 A1 **WO 2014/051558 A1**
US 20120054493 A1 **US 20100199095 A1**
Exalted, "WP5 - Security, Authentication &
Provisioning. Deliverable 5.1 Security and
Provisioning Solutions", (20120229)

(58) Field of Search:

As for published application 2518976 A viz:
INT CL **H04W**
Other: **WPI, EPODOC**
updated as appropriate

Additional Fields

INT CL **H04L**

Other: **None**

GB 2518976 B

Fig. 1

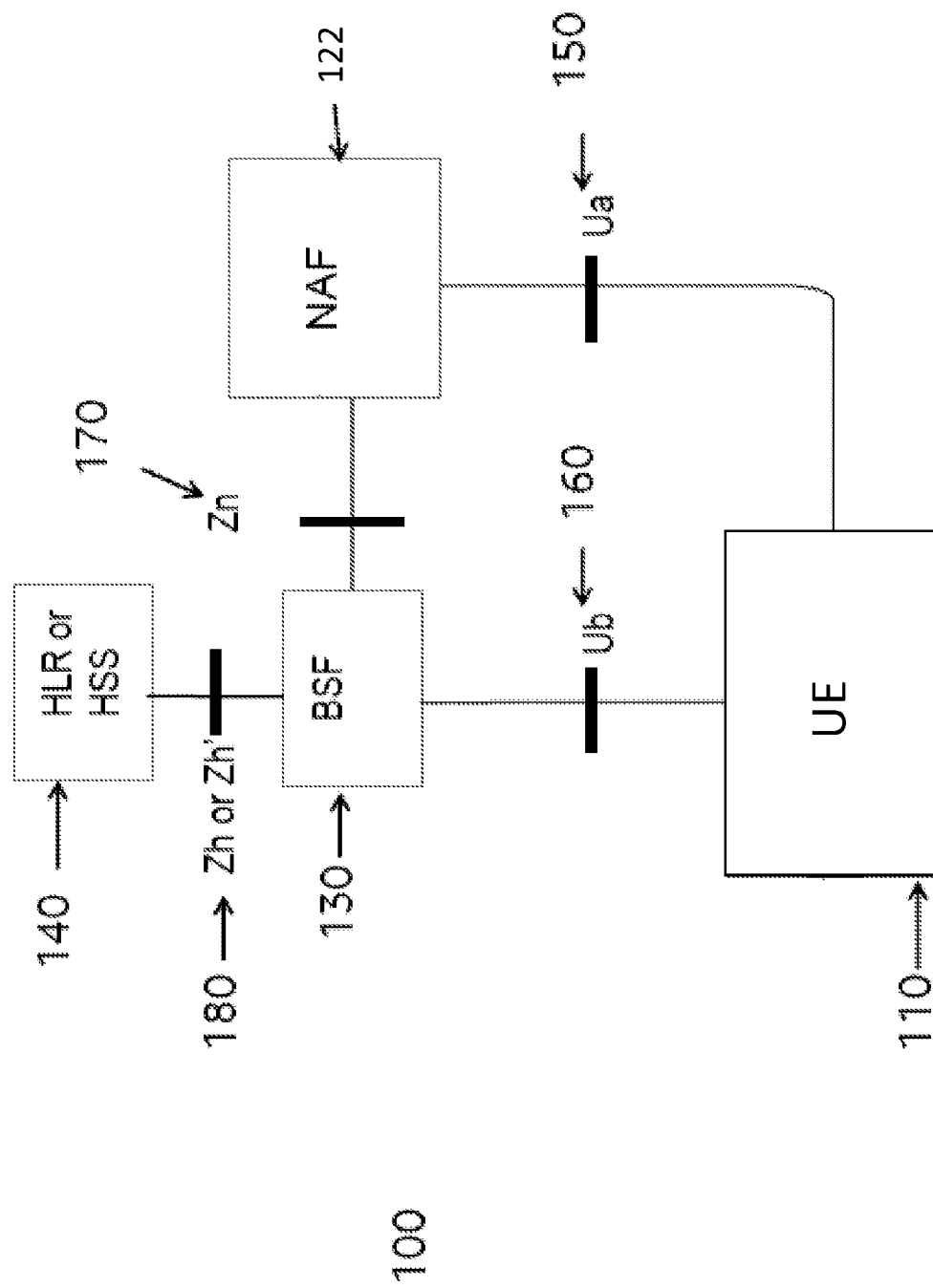


FIG. 1

Fig. 2

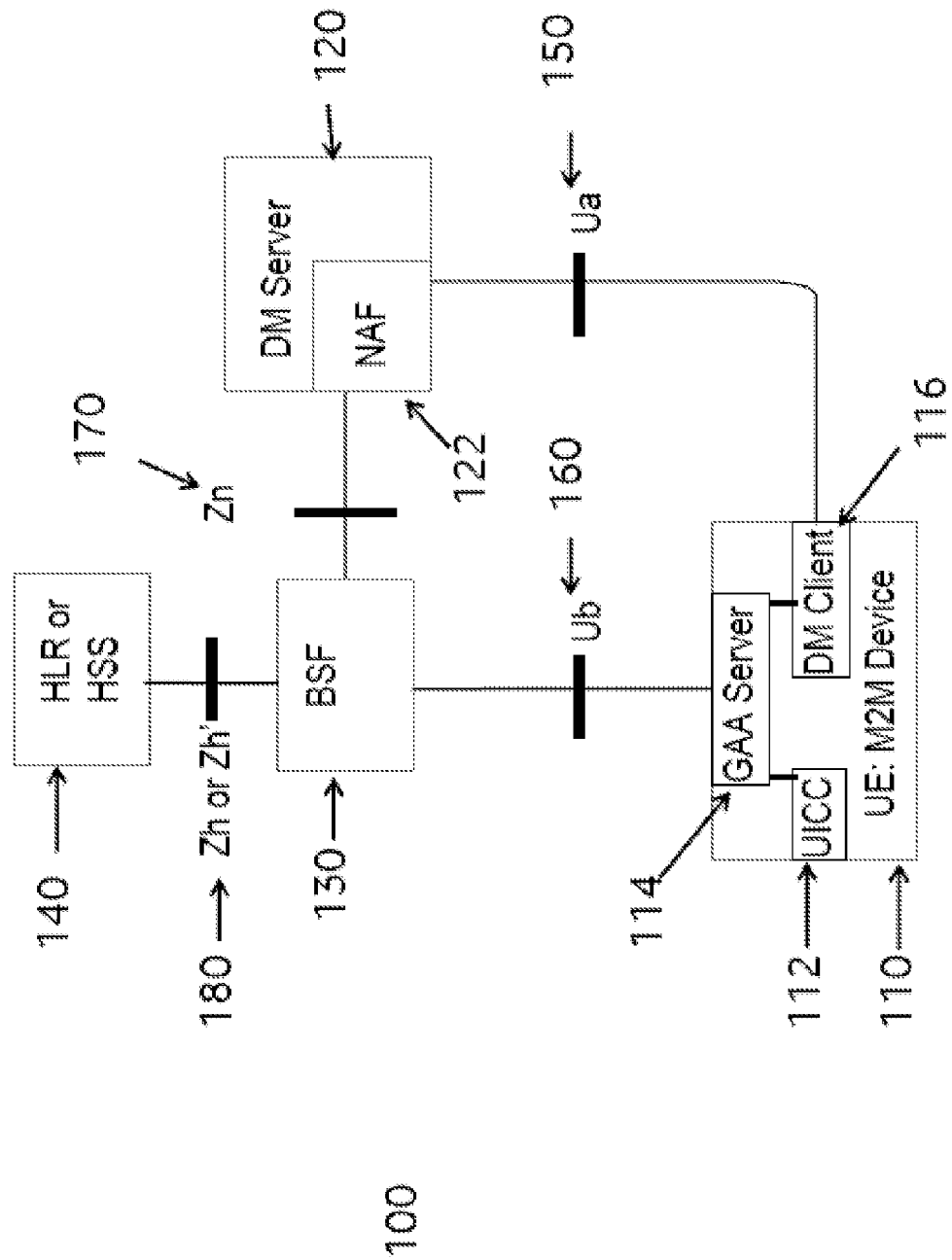


FIG. 2

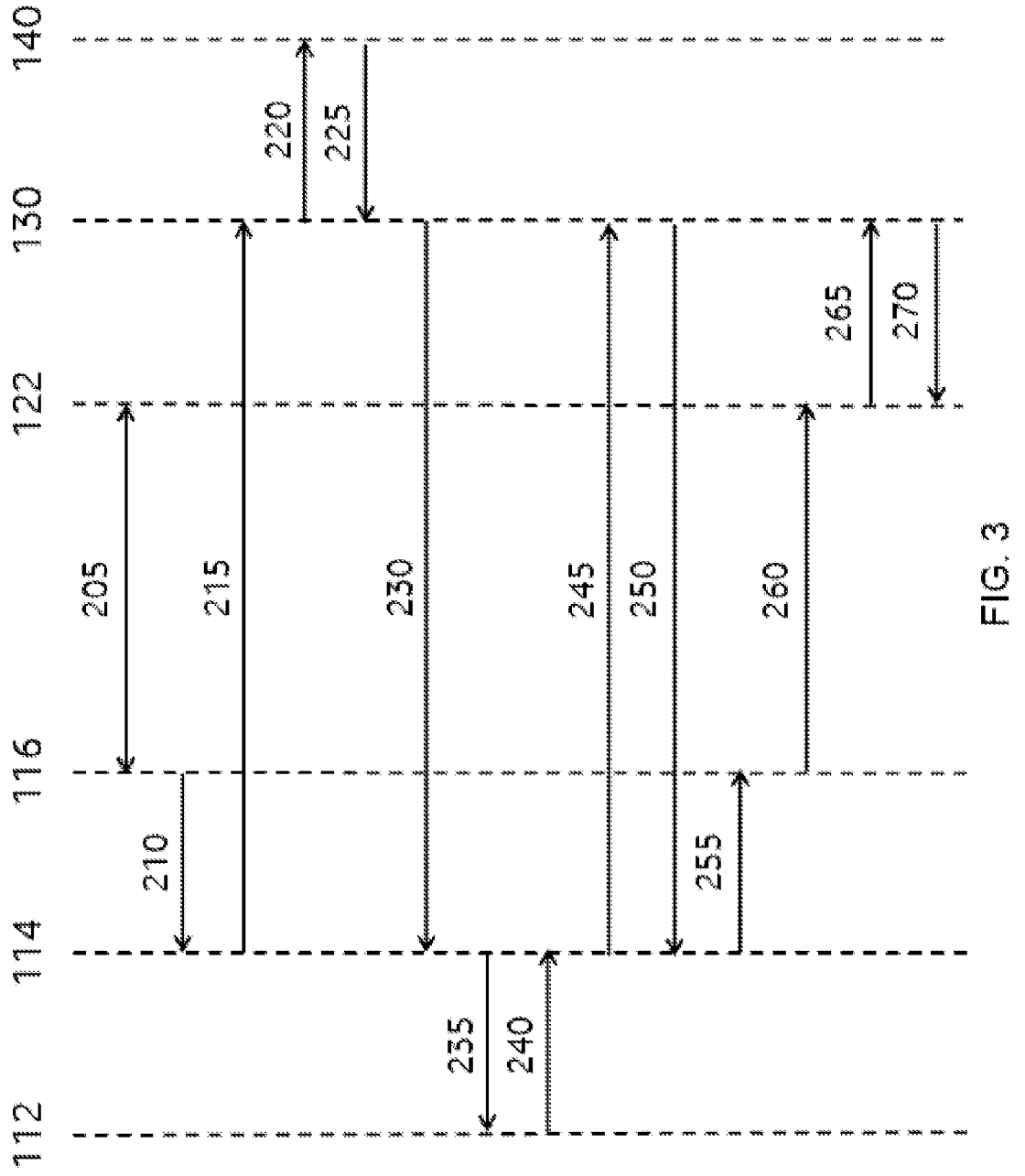
Fig. 3

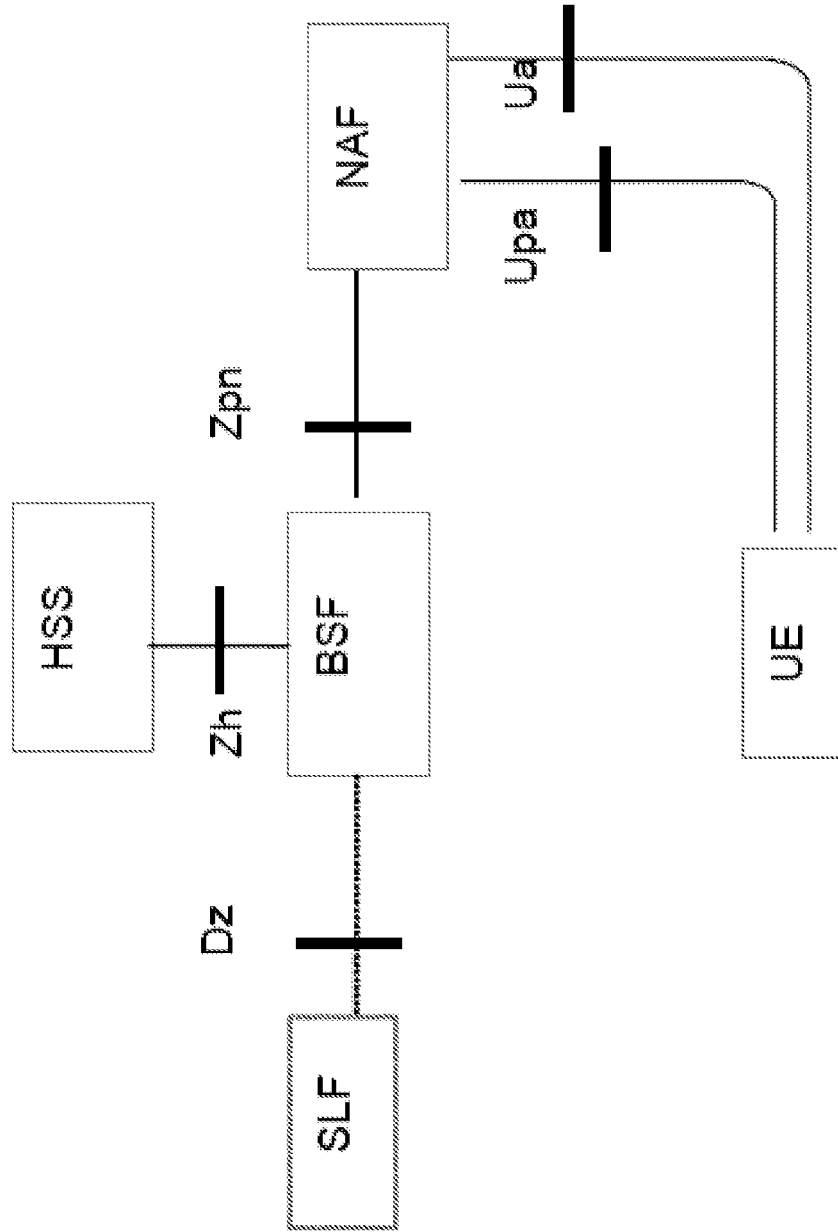
Fig. 4

FIG. 4

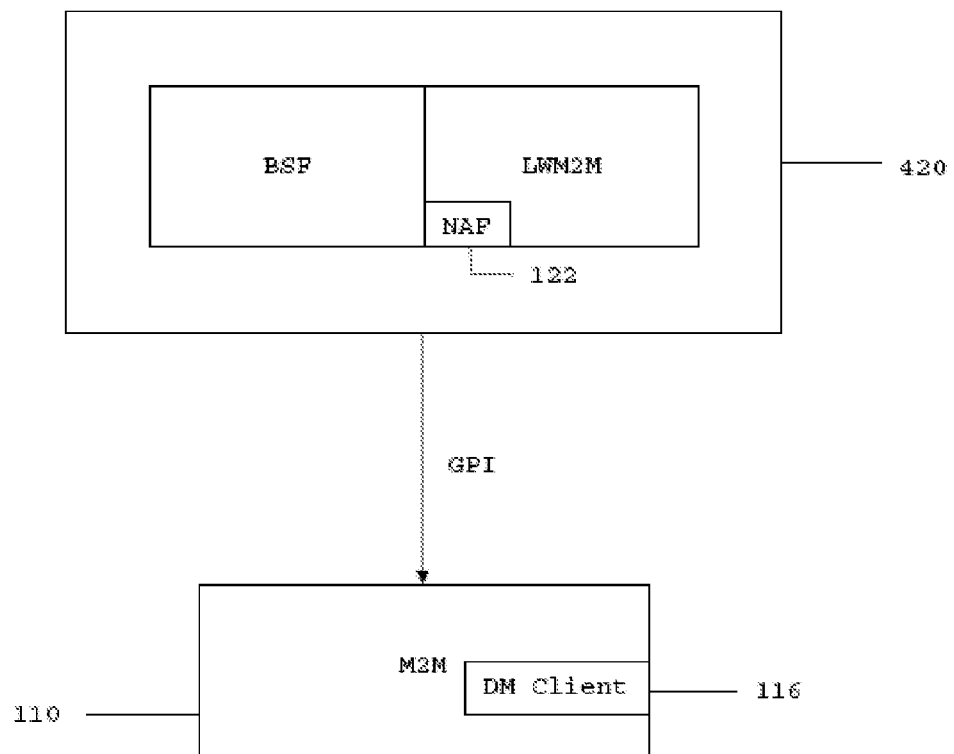
Fig. 5

Fig. 6

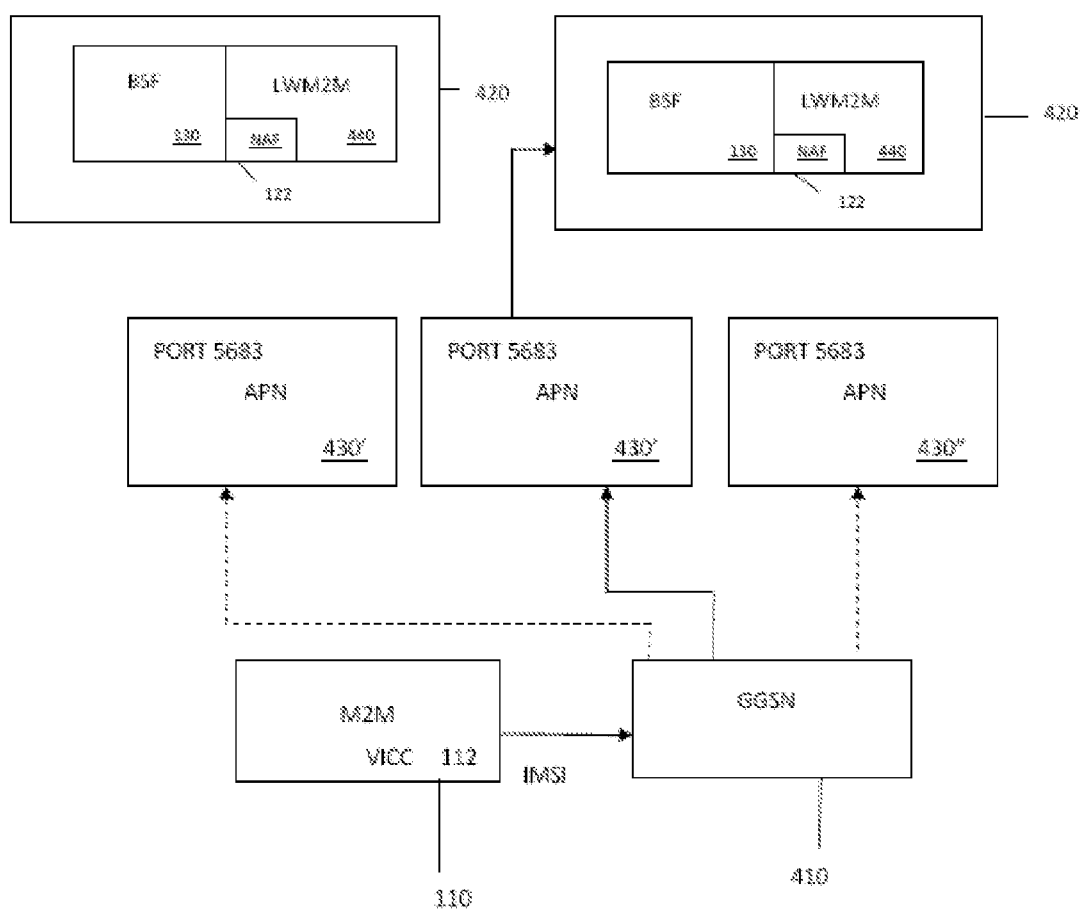


Fig. 7

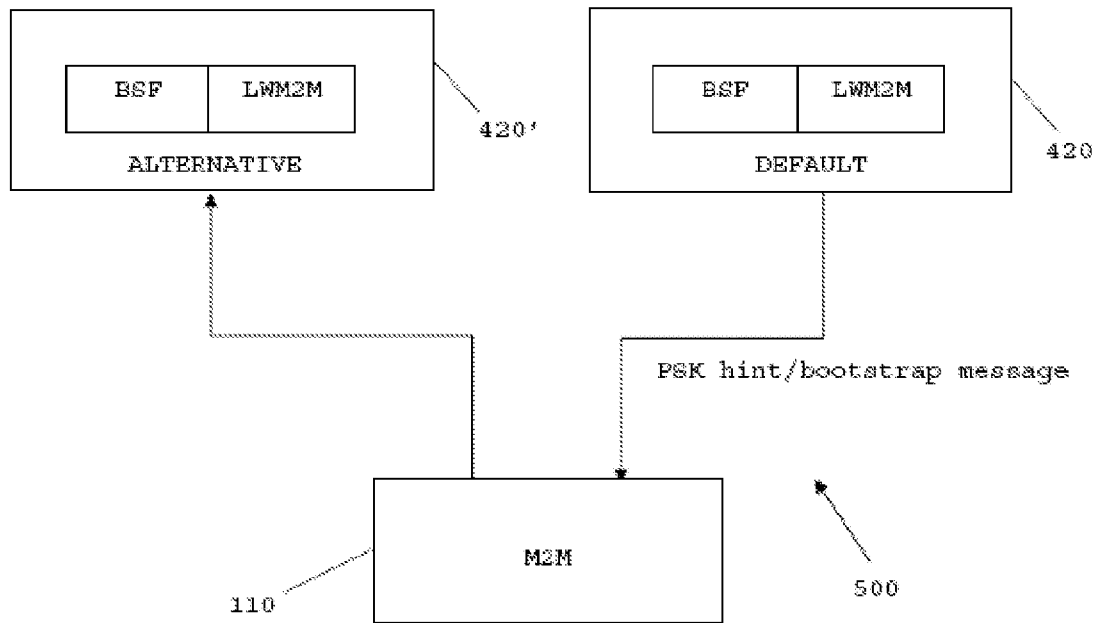


Fig. 8

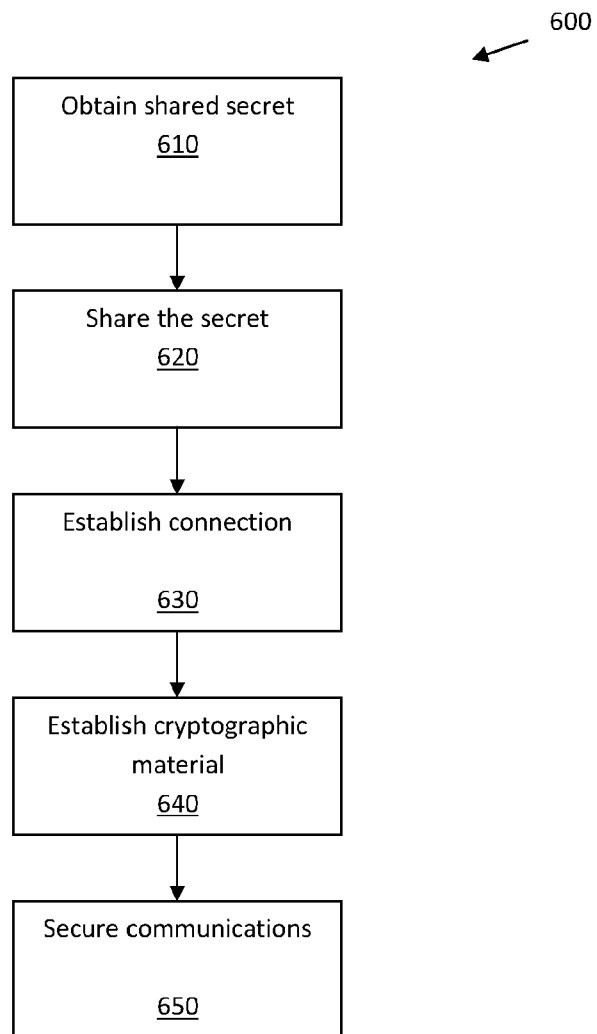
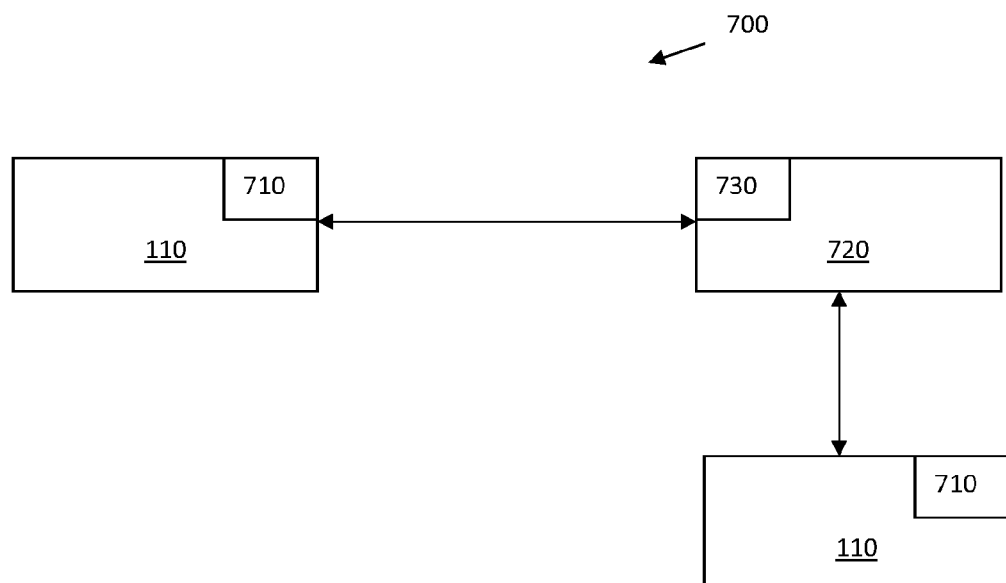


Fig. 9



SECURE COMMUNICATION WITH A MOBILE DEVICE

Field of the Invention

- 5 The present invention relates to a method and system for communicating securely with a mobile device and in particular a machine to machine (M2M) device.

Background of the Invention

- 10 Machine to Machine (M2M) devices are often numerous, hard-to-reach, and have constrained capabilities (owing to low cost, small size, low processing power or limited battery life). All of this makes their management, often remote, very complicated. Moreover, M2M devices often need to be managed in a secure manner. For example, they may contain information that is commercially sensitive and/or confidential for the one or
15 more entities that manage and/or own said devices. There is a need to remotely manage and communicate with them in a secure way, while respecting these constraints.

- One approach is to provide M2M devices with PKI keys for use in securing communications with a server or device manager. However, this increases complexity and requires such
20 the maintenance of such a provisioning system.

- Another approach is to use cryptographic material derived from an underlying key. However, if this key is ever compromised then an eavesdropper who has recorded previous communications can then decrypt these earlier messages.
25 Therefore, there is required a system and method that allows the M2M devices to communicate more reliably and more securely.

Details of 3GPP (RTM) standards and technologies used to implement aspects of the method and system

- 30 One of these architectures of 3GPP (RTM) is a Generic Authentication Architecture (GAA), which is a complex of standards which is described, for example, in 3GPP TS 33.919 (entitled "3G Security; Generic Authentication Architecture (GAA); System description", currently it may be retrieved at <http://www.3gpp.org/ftp/Specs/html-info/33919.htm>).

35

Generic Bootstrapping Architecture (GBA) is a 3GPP (RTM) standard defined in 3GPP TS 33.220 (entitled "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)", it could be currently retrieved at <http://www.3gpp.org/ftp/specs/html-info/33220.htm>). GBA is part of the complex of standards called GAA (see above).

5

GBA is a standard which enables a shared secret to be derived (bootstrapped) from the existing security association between a mobile network and a SIM card. This involves a network element called a Bootstrapping Server Function (BSF). In other words, GBA leverages the security of a SIM card (UICC) to authenticate mobile equipment, and then

10 derive key material for general-purpose applications.

15

GBA may be advantageously used to provide high-security to the communication between a client and the server, thus allowing remotely managing, controlling and, in general, communicating with a device in a high security manner. In particular, GBA (or a GBA-like architecture) is used for enabling a secure communication with the device (which, according to an aspect of the present disclosure, may be an M2M device), said communication being between a server and a client, the client being associated with the device, and wherein this communication is done for managing the device and/or services provided by (or via) the device, thus enabling a secure management of that device and/or

20 the services provided by (or via) the device. In this way, the device and/or the services provided by (or via) the device can be safely, securely and efficiently managed in a remote manner via a remote server.

25

GBA has been developed mainly for securing mobile broadcast (e.g. pay TV and equivalents). Indeed, standards for Multimedia Broadcast Multicast Service (MBMS) rely on GBA. Similarly, Open Mobile Alliance (OMA) Mobile Broadcast Services Enabler Suite (BCAST) smartcard profile relies on GBA. To date, most of the limited number of deployments of GBA in the world has been for mobile broadcast. GBA has also been standardised as an optional feature in conjunction with presence services, and within

30 miscellaneous "federated identity" services (e.g. Liberty Alliance, OpenID). In general, it is understood that GBA has been designed for use with mobile devices, such as mobile phones, laptop, computers, and many of the designed features have been provisioned with this in mind.

35

A variant of GBA, called "GBA Push", has been proposed for securing a message between a client and a DM server in the context of OMA Device Management Security. The OMA

27 07 20

Device Management is specifically designed for management of mobile devices such as mobile phones, tablet, computers, etc.

5 A different recent standard document (TS 102 690) merely mentions, in the context of M2M communications, the use of a standard GBA to secure communications between a device/gateway service layer and a network service layer.

10 There are some alternatives for identifying/authenticating a mobile user/device to a service. All of these alternatives are simpler than using GBA. For example, mobile operators and service providers can use WAP header enrichment.

15 Alternatively, the service provider can request the user to enter their phone number, send an SMS one-time password to that phone number, and ask the user to read the SMS and enter the password. These alternatives all work well with mobile devices and operators already, so service providers use them, although they are not as secure as GBA.

20 Additionally, many service providers prefer to offer services to a huge range of mobile devices, many of which do not contain a SIM card (e.g. PCs, laptops, Wi-fi-only tablets etc.). Since GBA relies on a SIM card/UICC in order to work, there has been no interest in using it.

25 Strong security is not possible with current alternatives such as a user-entered PIN or a bootstrapping message delivered by an SMS. These alternatives would either not be feasible or they would not provide the required level of security. First, there might not be a user around to enter a PIN (as most M2M devices operate independently from human intervention). Second, the service provider may be likely to want strong security (e.g. because M2M devices may include critical infrastructure), whereas PIN-based bootstrapping has weaker security. Third, if a PIN or SMS-based bootstrapping goes wrong (server connects to wrong client, client connects to wrong server, or there is a Man-In-The-Middle), then the user is likely to notice, complain and get it fixed, whereas an M2M device is unlikely to notice and complain, so may be permanently compromised. Neither is particularly practical by way of existing methods. For example, the OMA Device Management uses GBA Push for securing a message between a client and a DM server, and there is no explanation of how a similar architecture could be used or even modified for managing the device. Moreover, as mentioned above, the OMA Device Management is not compatible for use with an M2M device, as discussed above. This is particularly true for low

30

35

cost, simple M2M devices, such as simple sensors, switches, low cost trackers etc. Further, the standard document mentioned above uses a standard GBA to secure communications between a device/gateway service layer and a network service layer. Thus, the communication is not used for device/service management-related communications, and it is not clear, based on the observations made above, how a similar architecture could be used or even modified for managing the device from the server. Moreover, for the reasons mentioned above, the OMA Device Management and the standard document are incompatible, and a combination of the GBA Push for OMA Device Management with the standard document is not feasible, as it would result in the wrong device management protocol (i.e. one that is not suitable for M2M devices, particularly simple M2M devices), and some very laborious effort to make the two compatible and delete the elements which are redundant.

The OMA has defined a lightweight protocol for managing (as well as interacting with) M2M devices and managing services provided by M2M devices (e.g. remote control of attached sensors or machines). This protocol is called LWM2M, which is described in detail at http://technical.openmobilealliance.org/Technical/release_program/lightweightM2M_v1_0.aspx

This protocol runs over the CoAP protocol (analogous to http) – more specifically CoAP over DTLS (coaps) which is analogous to http over TLS (https). However, coaps requires a secure association to be provisioned between a device and a network server (DM Server) while providing no strong means to provision such an association from scratch.

A security aspect of OMA LWM2M is defined in Lightweight Machine to Machine Technical Specification Candidate Version 1.0 – 10 Dec 2013 (OMA-TS-LightweightM2M-V1_0-20131210-C).

In addition, there exists two protocols, the first one called DTLS defined in RFC 6347 (entitled “Datagram Transport Layer Security Version 1.2” ; it could be currently retrieved at <http://tools.ietf.org/html/rfc6347>); the second one called CoAP defined in draft-ietf-core-coap-18 (entitled “Constrained Application Protocol (CoAP) ”; it could be currently retrieved at <http://datatracker.ietf.org/doc/draft-ietf-core-coap/>). Both protocols are currently used in LWM2M. CoAP is still only an IETF draft (not a full RFC), and DTLS version 1.2 is also comparatively new (January 2012): versions of TLS have often existed as RFCs for several years before receiving widespread adoption.

The User Datagram Protocol (UDP) channel security for [COAP] is defined by the Datagram Transport Layer Security (DTLS) [RFC6347], which is the equivalent of TLS v1.2 [RFC5246] for HTTP and utilizes a subset of the Cipher Suites defined in TLS. (Refers to TLS Cipher Suite registry <http://www.iana.org/assignments/tls-parameters/tls-parameters.xml>) The DTLS binding for CoAP is defined in Section 9 of [CoAP]. DTLS is a long-lived session based security solution for UDP. It provides a secure handshake with session key generation, mutual authentication, data integrity and confidentiality.

The keying material used to secure the exchange of information within a DTLS session may be obtained using one of the bootstrap modes defined in Section 5.1.2 Bootstrap Modes of OMA LWM2M. The formats of the keying material carried in the LWM2M Security Object Instances are defined in Appendix E.1.1.

There also exists an authentication protocol HTTP Digest authentication, which is defined in RFC 3310 (entitled "Hypertext Transfer protocol (HTTP) Digest Authentication using Authentication and Key Agreement (AKA)", it can currently be retrieved at <http://www.ietf.org/rfc/rfc3310.txt>).

The GAA cluster of specifications TS 33.222 (entitled "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)") defines a general approach for pre-shared key TLS (TLS-PSK, RFC 4279). This can currently be retrieved at <http://www.3gpp.org/ftp/Specs/html-info/33222.htm>). For example, see especially Section 5.4.

In particular, with reference to GBA, 3GPP (RTM) Specification TS 33.220 defines the components and interfaces that are shown in figure 1. These are further described as:

NAF 122, the "Network Application Function" is a server-side component of an application that will be secured using GBA.

BSF, "Bootstrapping Server Function", 130 is a server-side component, which obtains authentication vectors from the HLR/HSS 140, and sends a challenge to the mobile device, "UE", 110 during the GBA protocol. On successful authentication, it derives the shared secret.

27 07 20

HLR/HSS 140, the "Home Location Register" or "Home Subscriber System", is the existing 3GPP (RTM) system which stores subscription details and credentials (the K and IMSI) for each SIM card (UICC) issued by a mobile operator. It may be

5 "GBA-aware" (so that it stores details for a GBA user subscription) or may be a legacy component.

UE, the "User Equipment", 110 is a mobile device containing a SIM card (UICC). The UE 110 supports a client application which communicates with the NAF 122, as well as a service which interfaces to the UICC, communicates with the BSF 130, and derives the
10 shared secret before passing it to the client application. This service is (somewhat confusingly) called a "GAA Server" in TR 33.905 (entitled "Recommendations for Trusted Open Platforms", it can currently be retrieved at <http://www.3gpp.org/ftp/specs/htmlinfo/33905.htm>).

15 Ua 150 is the interface between the Mobile Device (UE) 110 and the Network Application Function (NAF) 120.

Ub 160 is the interface between the Mobile Device (UE) 110 and the Bootstrapping Server Function (BSF) 130. This is specified in detail in TS 24.109 (entitled "Bootstrapping
20 interface (Ub) 160 and network application function interface (Ua) 150; Protocol details", it can currently be retrieved at <http://www.3gpp.org/ftp/Specs/html-info/24109.htm>).

Zh/Zh' 180 is the interface between the BSF 130 and the HSS or HLR 140. The Zh 180
25 interface is used with an HSS 140 that is "GBA Aware". The Zh' 180 interface is used with a legacy HLR or HSS 140. The Zh and Zh' 180 interfaces are specified in detail in TS 29.109 (entitled "Generic Authentication Architecture (GAA); Zh and Zn Interfaces based on Diameter protocol; Stage 3", it can currently be retrieved at <http://www.3gpp.org/ftp/Specs/html-info/29109.htm>) and TS 29.229 (entitled "Cx and Dx
30 interfaces based on the Diameter protocol; protocol details", it can currently be retrieved at <http://www.3gpp.org/ftp/Specs/html-info/29229.htm>).

Zn 170 is the interface between the NAF 122 and the BSF 130: this can use either a Web Services protocol (SOAP over http) or the Diameter protocol (RFC 3588). This is specified
35 in detail in TS 29.109 (see above).

27 07 20

There are a few other components and interfaces defined within the GAA standards, but these are not described in detail here.

5 There are several different versions of GBA defined in the standards. The flavours of GBA may include GBA-ME, GBA-U, GBA-SIM etc. The version called "GBA-ME" may require no special customizations of the UICC, except that the UICC does contain a 3G SIM (a USIM). However, other versions may be used. There may be a need to use the 2G variant of GBA (using a SIM rather than a USIM).

10 **Summary of the Invention**

In accordance with a first aspect there is provided a method as described in claim 1.

27 07 20
15 Therefore a connection or communication channel may be opened providing no or weak security or authentication and then used to start a security protocol allowing a greater level of security or authentication. For example, the connection may use an unsecure WiFi (RTM) hotspot. Whilst this connection doesn't have to be secure, it may be used to establish or authenticate material (e.g. keys) such that the cryptographic material is only made available to the M2M device and the server. The cryptographic material cannot be
20 recovered by any other party, even if that party knows the shared secret. This provides the perfect forward secrecy.

Optionally, the step of using the shared secret or data derived from the shared secret to establish the cryptographic material may further comprise the steps of:

25

30

35

the M2M device generating or obtaining a public key;
the server authenticating the public key of the M2M device using the shared secret
or data derived from the shared secret; and

the server deriving the cryptographic material from the public key of the M2M
5 device. The public key may have a corresponding private key. The M2M device may
provide a new public key, retrieve a stored public key or obtain the key from elsewhere, for
example.

Advantageously, the step of authenticating the public key of the M2M device may employ a
10 communication between a bootstrapping server and the M2M device. Or a communication
between the server and the bootstrapping server. For example the M2M device may pass
a public key or certificate to a BSF as a parameter within the bootstrapping protocol. The
BSF may pass the device's public key to a NAF as part of a GUSS or USS, for example.

15 Optionally, the step of using the shared secret or data derived from the shared secret to
establish the cryptographic material may further comprise the steps of:

the server generating or obtaining a public key;
the M2M device authenticating the public key of the server using the shared secret
or data derived from the shared secret; and
20 the M2M device deriving the cryptographic material from the public key of the
server. The public key may have a corresponding private key. The server may provide a
new public key, retrieve a stored public key or obtain the key from elsewhere, for example.

Advantageously, the step of authenticating the public key of the server may employ
25 a communication between a bootstrapping server and the M2M device, or a communication
between the server and the bootstrapping server. For example, the BSF may pass the
public key or certificate of the server (or an authority public key used to sign the server's
certificate) to the M2M device as a parameter within the bootstrapping protocol. The BSF
may create a certificate for the NAF, which the NAF can use when communicating with the
30 device, for example.

Advantageously, the step of using the shared secret or data derived from the shared secret
to establish the cryptographic material on both the M2M device and the server may further
comprise the steps of:

35 the M2M device generating or obtaining a public key and a corresponding private
key;

the server generating or obtaining a public key and a corresponding private key;
the M2M device authenticating the public key of the server using the shared secret
or data derived from the shared secret;

the server authenticating the public key of the M2M device using the shared secret
5 or data derived from the shared secret;

the M2M device deriving the cryptographic material from the public key of the server
and the private key of the M2M device; and

the server deriving the cryptographic material from the public key of the M2M device and
the private key of the server. In this embodiment, the M2M device and server each
10 generate a public/private key-pair, each use the shared secret to authenticate the other's
public key. Each uses its own private key and the other's public key to derive the
cryptographic material. This is one mechanism that prevents the cryptographic material
from being derived or obtained from simply knowing the public keys and the original shared
secret.

15

Preferably, the cryptographic material may be session keys to provide perfect forward
secrecy. This means that if the secret is compromised then old or historic messages are
not necessarily subject to unauthorised recovery.

20 Preferably, the cryptographic protocol may be TLS, SSL, or datagram transport layer
security, DTLS. These may take the form of an Elliptic curve Diffie-Hellman key agreement
protocol, for example.

Optionally, the shared secret may be generated by or stored within the M2M device and the
25 shared secret or data derived from the shared secret are sent to the server. The shared
secret may be pre-provisioned or generated on first power-up, for example.

Optionally, the method may further comprise the step of the server authenticating the
shared secret. The server may in some embodiments, authenticate the shared secret (or
30 derived material).

Optionally, the shared secret may be generated by or stored within the server and the
shared secret or data derived from the shared secret are sent to the M2M device.

Optionally, the method may further comprise the step of the M2M device authenticating the shared secret. Optionally, both the M2M device and the server may authenticate shared secret (or derived material) generated or stored on the other entity.

- 5 Optionally, the data derived from the shared secret may be obtained by applying a hash function or other fixed key derivation function to the shared secret. Other derivation techniques may be used.

10 Optionally, the server may be a device management server, device management bootstrap server, a lightweight M2M, LWM2M, server, a LWM2M bootstrap server or a Network Application Function, NAF or other entity.

Optionally, the server may be a Bootstrapping Server Function, BSF.

- 15 Optionally, the method may further comprise the step of authenticating the shared secret or using the shared secret to authenticate either or both the M2M device and the server.

Advantageously, the connection established between the M2M device and the server may be unsecured or be only weakly secured.

20 Advantageously, the secured communication between the M2M device and the server is over the connection. In some embodiments the secured communication is over a separate connection.

- 25 Preferably, the step of securing the connection between the M2M device and the server further comprises providing Generic Bootstrapping Architecture, GBA, or GBA Push Info, GPI, messages over the connection or over a separate connection.

30 Optionally, the shared secret may be used to establish, protect, endorse or authenticate any one or more of a certificate, digital certificate, a self-signed certificate, or a certificate signed by an unknown or unverifiable certificate authority. Other material may be used. The private key may be held by the M2M device. The private key may be held by the server. Both the M2M device and the server may each hold a private key, and each corresponding public key or certificate is established, protected, endorsed or authenticated
35 using the shared secret, for example.

Optionally, the shared secret may be used to establish, protect, endorse or authenticate an authority key or certificate. The authority key is then used to sign, endorse or authenticate a public key or certificate for the device and/or server. For example, the shared secret may be used to distribute a certificate authority (CA) certificate for a previously unknown CA.

5

Optionally, the shared secret may be used to verify that a public key or certificate is still valid (e.g. has not expired or been revoked), including an authority key or certificate. Or the shared secret may be used to authenticate an updated public key or certificate, including an authority key or certificate. For example, the shared secret may be used like this where

10 the device does not have access to an accurate clock (so cannot check expiry time), or where the device cannot fetch revocation information like CRLs, or the shared secret may be used to update or replace CA certificates.

Preferably, the shared secret may be configured to expire at a predetermined time.

15

Therefore, there is no need to actively expire material.

Optionally, the server may generate a transaction identifier configured to expire at a predetermined time and further wherein the expiry time of the shared secret corresponds with the expiry time of the transaction identifier.

20

Advantageously, the method may further comprise the step of obtaining information indicating current time from a trusted time source. Therefore, this may be relied upon to determine whether or not the shared secret has expired.

25

Preferably, the information indicating the current time may be obtained from messages from the trusted time source secured using GBA.

According to a second aspect, there is provided a system as described in claim 20.

30

Optionally, the one or more M2M devices may be configured to generate or obtain a public key and a corresponding private key, and wherein the server is further configured to generate or obtain a public key and a corresponding private key;

the one or more M2M devices are configured to authenticate the public key of the server using the shared secret or data derived from the shared secret;

35

the server is configured to authenticate the public key of the one or more M2M devices using the shared secret or data derived from the shared secret;

29 04 20

the one or more M2M devices are configured to derive the cryptographic material from the public key of the server and the private key of the M2M device; and
the server is configured to derive the cryptographic material from the public key of the M2M device and the private key of the server.

5

The methods described above may be implemented as a computer program comprising program instructions to operate a computer. The computer program may be stored on a computer-readable medium.

10

The computer system may include a processor such as a central processing unit (CPU). The processor may execute logic in the form of a software program. The computer system may include a memory including volatile and non-volatile storage medium. A computer-readable medium may be included to store the logic or program instructions. The different parts of the system may be connected using a network (e.g. wireless networks and wired networks). The computer system may include one or more interfaces. The computer system may contain a suitable operation system such as UNIX, Windows (RTM) or Linux, for example.

15

It should be noted that any feature described above may be used with any particular aspect or embodiment of the invention.

20

Brief description of the Figures

29 04 20

The present invention may be put into practice in a number of ways and embodiments will now be described by way of example only and with reference to the accompanying drawings, in which:

Figure 1 shows a schematic diagram of components and interfaces with which GBA may
5 be used;

Figure 2 shows a schematic diagram of an example of an architecture that can be used in accordance with the present invention, in particular when GBA is used;

Figure 3 shows an exemplary flow diagram of communications exchanged within the exemplary architecture of Fig.2;

10 Figure 4 shows a schematic diagram of an example of an alternative architecture that can be used in accordance with the present invention, in particular when generic bootstrapping architecture (GBA) is used;

Figure 5 shows a schematic diagram of a device management (DM) server in communication with a machine to machine (M2M) device;

15 Figure 6 shows a schematic diagram of a system and method for initiating communications between the DM server and M2M device of figure 5;

Figure 7 shows a schematic diagram of a further system and method for initiating communications between the DM server and M2M device of figure 5;

Figure 8 shows a flowchart of a method for securely communicating with an M2M device;
20 and

Figure 9 shows a schematic diagram of a system for implementing the method of figure 8.

It should be noted that the figures are illustrated for simplicity and are not necessarily drawn to scale. Like features are provided with the same reference numerals.

25

Detailed description of the preferred embodiments

A device may communicate securely with a server. The device may be a Machine to Machine (M2M) device, or an equivalent device (e.g. a device, a generic or specific
30 communication device, including one or more modules capable of providing M2M capabilities).

Aspects of the Generic Authentication Architecture (GAA) and Generic Bootstrapping Architecture (GBA) are identified in "Details of 3GPP standards and technologies used to
35 implement aspects of the method and system" above. In particular, the specific architecture on which the method and system may be based is GBA.

Generic Bootstrapping Architecture (GBA) uses existing security associations between a network (e.g. a mobile network) and a card (e.g. a SIM card or UICC) to derive a key that can be used for the secure communication between the client and the server. Accordingly, if the device is associated with such a card, as well as with the client, the method can advantageously use the GBA to derive the security elements (e.g. a shared secret) to enable the client associated with the device to securely communicate with the server. Accordingly, the device could be advantageously adapted so that it is associated with the card and the client and uses GBA to derive the security elements for secure communication with the server. Moreover, as GBA is standards-based, the impact of the required modifications may be relatively limited and the overall solution would be very attractive (in particular, to M2M users/owners as well as to network operators and/or service providers).

M2M devices are different from the mobile devices that OMA Device Management was originally designed for (such as mobile phones, laptops, computers, as explained in "Details of 3GPP standards and technologies used to implement aspects of the method and system" above), and use of GBA (in any of its versions) with M2M is not a straightforward implementation.

A variant of GBA, called "GBA Push" has been proposed for securing a message between a client and a DM server in the context of OMA Device Management Security, and is identified in "Details of 3GPP standards and technologies used to implement aspects of the method and system" above. It is noted that, although GBA Push and GBA are related, it is not trivial to use GBA in place of GBA Push (and vice versa). This is because these two architectures have some important differences. First, in GBA the device has to contact the BSF in order to request a RAND and AUTN (and use this to derive a Ks_local). To the contrary, in GBA Push, the client does not have to talk to the BSF - it just receives a message prepared by the BSF. Furthermore, in GBA, there is no need to modify the Ua interface. In GBA Push, either the Ua interface has to be modified in some way to carry the push message or a new interface must be added. Accordingly, GBA Push cannot be used with an arbitrary application protocol. For GBA Push, the application protocol has to be "GBA aware" in some sense (e.g. so it can carry the GBA Push Info (GPI) messages). In GBA, the Ks_local can be used to derive several different Ks_NAFs (e.g. for different application servers). In GBA Push, only one NAF can use/rely on the Ks_local. Accordingly, GBA Push is slightly less efficient than GBA.

The GBA Push Info is described in 3GPP TS 33.223, Section 5.2.1. The encoding is defined in Section 5.3.5. See in particular table 5.2.1.1 and figure 5.3.5.1 in 3GPP TS 33.223 V12.0.0 (2013-12) that may be found:

5 http://www.3gpp.org/ftp/Specs/archive/33_series/33.223/33223-c00.zip

Moreover, as discussed above, M2M devices are very constrained in their capabilities (e.g. computation, communication, life, etc.) and these constraints make their management more complex and harder to implement in a simple manner. GBA requires a number of
10 interfaces and components which are hard to implement with M2M (for examples and description of these interfaces and components, please refer to the sections below).

In order to more efficiently and securely manage the device and/or services provided by (or via) the device, these interfaces and components need to be modified or otherwise adapted
15 so they can properly and effectively work with M2M devices.

For example, carrying the Ub interface (and associated protocol) over constrained M2M devices is very difficult. For example, the standard Ub interface uses HTTP and HTTP digest. The likely rationale for this is that, as mentioned above, GBA was designed having
20 mobile devices, such as mobile phones, in mind. So, since all phones use HTTP, and therefore all have an HTTP stack, then HTTP was the easiest protocol to be used for the Ub interface. However, this is not true for M2M devices. For example, according to the Lightweight M2M (LWM2M) protocol (see below for more details), a protocol called CoAP is used in M2M devices, precisely because it is a simpler/more efficient alternative to HTTP.
25 Alternatively, this Ub interface could be tunnelled, for example via another interface (e.g. the Ua), so that the system may be simplified.

Additionally, building all the necessary components (e.g. GAA server, interfaces) into a capacity-constrained M2M device appears to be very difficult. For example, physical and
30 virtual space constraints, as well as computational constraints, create considerable problems for building the necessary components. Moreover, having one or more interfaces between M2M application(s) and a card on the device, such as a UICC, is very difficult. This is due, for example, to the fact that most M2M modems do not support the required low level interface(s). In general, the overall integration of the GBA required interfaces and
35 components with an M2M device appear very difficult. A possible, but not optimal solution, could be to pre-provision the M2M devices (e.g. having the M2M devices already designed

and/or manufactured with the required components and interfaces) and the associated elements required for use of GBA (e.g. the card being capable of interfacing with the M2M device) so that the GBA could be used. To date, no M2M device is pre-provisioned with these characteristic.

5

In addition, as noted above, GBA is not widely used. There are other reasons why GBA is not widely used. For example, use of GBA requires support in the device, in the network (e.g. BSF – see below) and by individual services (which may be deployed, for example, by a mobile operator or by other parties). In the preferred use-case (mobile broadcast) support is also required in the SIM card (as it uses GBA-U). Accordingly, a lack of coordination and willingness to act/cooperate between the various parties involved in this deployment (e.g. device manufacturers, mobile operators, service providers) has so far blocked implementation of GBA.

10

15

For all the above reasons, GBA (or a GBA-like architecture, for example a variant and/or a suitably modified version) may be used for enabling a secure communication with a device (in particular, an M2M device). The communication may be between a server and a client, the client being associated with the device, and wherein this communication may be done for managing the device and/or services provided by (or via) the device. This enables a secure management of that device and/or the services provided by (or via) the device and creates a new and innovative combination which produces a synergistic effect and provides many technical advantages.

20

For instance, as already mentioned above, the GBA will provide a higher and very strong level of security to the device/service management-related communications with M2M devices, which is a very critical and important point.

25

Another advantage, in addition or combined with the strong security described above, is in terms of full automation. Moreover, an M2M service provider does not have the cost/complexity of setting up their own security solutions, as the solution can be provided directly by the mobile operator implementing the solution described in this application. In particular, a service provider does not have to set up a PKI, issue certificates, pre-load keys to devices and so on.

30

Accordingly, the method may further comprise that the provision of the secure communication is based on a security association between a network and a card, the card

35

being associated with the device. For example, the card may be embedded within the device (e.g. soldered in the device) or provided to the device by way of a suitable connection. In general, the card may be associated in any suitable manner so that there is an association between the card and the device. The network can be a mobile network, or
5 any equivalent network, while the card can be a SIM card, a UICC, or any card associated with the network. The method may further comprise deriving a shared secret based on the security association. The method may further comprise providing the client and the server with the shared secret so as to enable the secure communication. The server may be a server adapted to manage the device (e.g. remotely manage the device, send updates,
10 transfer information to and from the device, control device parameters, etc.) and to manage services provided by the device (e.g. device is used to switch on/off and/or dim streetlights). The shared secret may be a key and/or a similar security arrangement.

The method may further comprise authentication between the client and the server. The
15 authentication may be based on the shared secret. The authentication may be performed via an authentication component. The authentication may be performed by means of a first authentication between the client and an authentication component and of a second authentication between the server and the authentication component. The client and the server may be independently authenticated by an authentication component. As a result of
20 the client and the server being authenticated by the authentication component, both the client and the server may share the shared secret. The authentication may be performed by means of the shared secret. The shared secret may be shared between the client and the server. Alternatively, the shared secret may be shared between the client, the server and the authentication component. The authentication may implicitly result from the client,
25 the server and the authentication component sharing the shared secret. The method may further comprise deriving a second shared secret based on the shared secret, the second shared secret being shared between the client and the server. This second shared secret may then be used for the authentication as discussed above.

30 The obtainment of the shared secret at the client may be based on an identifier associated with a server authentication component. The shared secret may be obtained at the server from the authentication component. The obtainment of the shared secret at the server is obtained based on an identifier associated with the shared secret. The identifier is generated by the authentication component. The identifier may be provided to the server by
35 the client.

The OMA LWM2M protocol for managing (as well as interacting with) M2M devices and managing services provided by M2M devices (as described in "Details of 3GPP standards and technologies used to implement aspects of the method and system") may be used. However, other device management protocols may be used or the method and system may
5 be extended to other M2M services (for example, securing the delivery of binary SMS).

GBA could be advantageously used in conjunction with LWM2M in order, for example, to establish keys for LWM2M, whilst at the same time LWM2M and the procedures specified therein could be used to transport and/or carry any message and/or communication which
10 relates to GBA. For example, this can be done by using specific tunnels (e.g. Ub) or GBA Push Info (GPI) messages. The use of GBA together with LWM2M creates a new and innovative combination which produces a synergistic effect and provides many technical advantages. For example, it allows addressing many more low-end devices, such as M2M devices. This is due, for example, to the use of a device management protocol which is
15 properly optimized for M2M, rather than one repurposed from the consumer space (e.g. OMA DM v1, TR-069). This optimised protocol can be used to transport GBA messages – avoiding the need for a separate HTTP stack – and to manage GBA parameters (identifiers for device and application, lifetimes, key derivation methods, etc.). Further, when accompanied by appropriate network systems to provide automated routing and discovery
20 (e.g. of LWM2M server and BSF), GBA and LWM2M advantageously combine to eliminate the cost of pre-loading settings and credentials, so facilitating low cost devices. GBA with LWM2M securely supports low-cost devices which are unattended or have no UI, where there is no option for user interaction (such as entry of PIN), and where there is no user who is able to notice and recover from authentication failures (spoof server, spoof client or
25 Man In The Middle). Moreover, GBA works without requiring any public key or certificate processing on the device. This is particularly advantageous on simpler devices, as these devices may have minimal public key support or implementation errors when handling certificates.

30 Accordingly, the shared secret may be used as a key in the LWM2M standard. Also, the LWM2M standard procedures may be used for transmission and/or reception of any communication used within the GBA.

The shared secret may be used as a key or shared secret within the DTLS protocol
35 (identified in "Details of 3GPP standards and technologies used to implement aspects of the method and system" above), either when the LWM2M is used in conjunction with a

DTLS protocol or when the DTLS is used alone or in conjunction with one or more other protocols.

5 The secure communication may further be a data communication. The data communication may be an SMS-based communication. An SMS binding may be used. The data communication may be a UDP-based communication.

10 The method may further comprise encrypting a communication over the secure data communication. The encryption may be performed using an Advanced Encryption Standard. The SMS-based communication may be further secured by use of an Over-The-Air (OTA) protocol, e.g. a Secured Packet Structure for UICC Applications. This protocol is defined in ETSI standard 102.225. The OTA protocol may be arranged to secure the communication with the identification card associated with the device.

15 It has also been noted that the OTA protocol can be used advantageously in conjunction with the LWM2M standard, in which the LWM2M can be used to manage parameters, keys and similar elements for the OTA protocol.

20 The use of OTA with LWM2M is not a straightforward implementation. OTA is a solution designed for SIM card security, as it has some real technical challenges if used for LWM2M. In particular, while there is software written for SIM cards and SIM OTA servers to support ETSI standard 102.225, a similar software does not exist in the device management space for devices (and, in particular, not for OMA DM clients and servers). Thus, M2M device manufacturers do not have a code-base that they can easily adapt for
25 use with these devices.

Further, the ETSI standard 102.225 does not explain how to set up the keys and parameters for use with the standard. It simply assumes the keys and parameters are all pre-loaded and known to both SIM card and OTA server. Although this assumption is
30 acceptable in the SIM space - because SIM cards can be securely provisioned with the necessary keys at the manufacturing stage, and SIM manufacturers have interfaces with operators for communicating the necessary keys and parameters – the same cannot be said about LWM2M, where that infrastructure does not exist.

35 Thus, the use of OTA together with LWM2M creates a new and innovative combination which produces a synergistic effect and provides many technical advantages. For example,

the SMS bearer needs to be secured, and so far no solution has been found. Use of OTA enables the SMS bearer to be used in LWM2M. Without it, it would not be possible to use SMS-based communications in LWM2M, and that would limit the applicability of the overall LWM2M standard.

5

Accordingly, the LWM2M standard procedures may be used to manage parameters and/or keys used in the OTA protocol. The method may further be used in conjunction with LWM2M, as described above.

10 It has also been noted that the method described above, implemented using the GBA (or a similar architecture), can be used in conjunction with SMS so that the GBA can be employed to establish keys for secure SMS-based communications (e.g. SMS), while at the same time SMS-based communications can be used to transport or carry messages associated with GBA – for example, carry GBA Push Info (GPI) messages. The use of
15 SMS-based communications together with GBA creates a new and innovative combination which produces a synergistic effect and provides many technical advantages. For example, GBA can be used to establish the shared keys that are needed to protect SMS, while using SMS as a transport to deliver the necessary GBA messages. Further the SMS used to deliver the GBA messages can themselves be integrity protected (and partly encrypted)
20 using the keys that will be established by GBA, so at no point is there a reliance on not secure SMS. This synergistic combination allow use of SMS as the sole bearer for M2M traffic, something which would not otherwise be possible, except by preloading the keys needed to secure SMS traffic, or switching to a different protocol to negotiate these keys: both of these alternatives would add complexity and cost. Thus, it would provide a very
25 high security solution for obtaining shared keys so that the security of the keys is not compromised, and at the same time an-SMS-based communication is enabled by virtue of the provisioning of the keys.

Accordingly, when the method is implemented using GBA, the GBA may be used to
30 establish keys for secure transmission and/or delivery of SMS. SMS-based communications may be used for transmission and/or reception of any communication used within the GBA, noting that these communications may themselves be protected using the keys that will be derived in GBA.

35 In addition to the above, the server may further comprise a server authentication component. Also, the client may further comprise a client authentication component. The

server authentication component may perform authentication of the server with the authentication component. The client authentication component may perform authentication of the client with the authentication component.

- 5 Further, the authentication component may be a Bootstrapping Server Function (BSF), the server authentication component may be a Network Application Function (NAF) and the client authentication component may be a GAA Server.

10 The method may further comprise communicating between the server and the client for determining security parameters to be used for the secure communication, wherein the communicating is performed by using a device management protocol (for example, the GBA). The secure communication may be for use in the device management protocol.

15 In a further embodiment, there is provided a method of enabling secure communication for use in a device and/or service/application management protocol, the secure communication being between a server and a client, the client being associated with a device, the secure communication requiring security parameters to be agreed between the client and server, the method comprising communicating between the server and client to agree the security parameters, wherein the communicating is performed by using the
20 device management protocol. The device can be an M2M device.

In a further embodiment, there is provided an apparatus, system, module or network for enabling secure communication with a device, said communication being between a server and a client, the client being associated with the device. In addition, the apparatus, system,
25 module or network may further include means for performing any one of the steps or features of the methods described above. The device can be an M2M device.

In a further embodiment, there is provided an apparatus, system, module or network for enabling secure communication for use in a device and/or service/application management
30 protocol, the secure communication being between a server and a client, the client being associated with a device, the secure communication requiring security parameters to be agreed between the client and server, the method comprising communicating between the server and client to agree the security parameters, wherein the communicating is performed by using the device management protocol. In addition, the apparatus, system,
35 module or network may further include means for performing any one of the steps or features of the methods described above. The device can be an M2M device.

In a further embodiment, there is provided a client including any means, features or functionalities corresponding to the means, features or functionalities relative to the client as recited by any one of the methods described above.

5

In a further embodiment, there is provided a server including any means, features or functionalities corresponding to the means, features or functionalities relative to the server as recited by any one of the methods described above.

10 In a further embodiment, there is provided a device comprising a card and a client, wherein the device is arranged for enablement of secure communication, the secure communication being between a server and the client, wherein the provision of the secure communication is based on a security association between a network and the card. The client may comprise any means, features or functionalities corresponding to the means, features or
15 functionalities relative to the client as recited by any one of the methods described above. The device can be an M2M device.

In a further embodiment, there is provided a server arranged for enablement of secure communication with a device, the secure communication being between the server and a
20 client associated with the device, wherein the provision of the secure communication is based on a security association between a network and a card, the card being associated with the device. The server may comprise any means, features or functionalities corresponding to the means, features or functionalities relative to the server as recited by any one of the methods described above. The device can be an M2M device.

25

In a further embodiment, there is provided a system for enabling secure communication with a device, said communication being between a server and a client, the client being associated with the device, wherein the provision of the secure communication is based on a security association between a network and a card, the card being associated with the
30 device. The device can be an M2M device.

In a further embodiment, there is provided a method of enabling secure data communication with a device, the communication being between a server and a client associated with the device, wherein the security of the communication is enabled by a
35 bootstrapped secret. The device can be an M2M device. The security protocol may be used to secure the data communication. The bootstrapped secret may be used to obtain

the security elements used for the secure protocol. The bootstrapped secret may be a pre-shared secret, said secret being directly provided to the server and the client. The pre-shared secret may be permanently provided to the server and the client (e.g. by pre-provisioning the client and/or the server with said pre-shared secret, e.g. at manufacturing stage or before the client and/or server are used in a system). The pre-shared secret may be a strong, high entropy or a temporary, low-entropy pre-shared secret. The bootstrapped secret may be based on a public key or a certificate-based method. The bootstrapped secret may be provided via a bootstrap server. The security elements can be keys and/or similar arrangements well known in the art.

The communication may be an SMS-based communication. The security protocol is defined by ETSI TS 102.225. The method may use SMS binding. The device may be further associated with a card, and the security of the data communication may be controlled by means of the card. Any incoming SMS-based communication may be decrypted and/or checked by means of the card, and/or any outgoing SMS-based communication may be encrypted and/or checked by means of the card.

The communication may be a UDP-based communication. The security protocol may be a DTLS protocol.

The secure data communication may be provided over a communication interface. The communication interface may be used for managing the device or for managing the bootstrapping operations.

The data communication may be performed according to the LWM2M protocol.

In a further embodiment, there is provided an apparatus, system, module or network for enabling secure data communication with a device, the communication being between a server and a client associated with the device, wherein the security of the communication is enabled by a bootstrapped secret. The device can be an M2M device.

In a further embodiment, there is provided a method of retrieving security elements required for enabling secure data communication with a device, the communication being between a server and a client associated with the device, wherein the security elements are retrieved using a bootstrapping protocol. The device can be an M2M device. The bootstrapping protocol may retrieve the security elements in a secure session. The session

may be secured based on a security protocol. The security protocol may be a DTLS protocol. The bootstrapping protocol may be based on GBA. The data communication may be an SMS-based communication. The bootstrapping protocol may be a LWM2M bootstrap protocol. The security elements can be keys and/or similar arrangements well known in the art.

In a further embodiment, there is provided an apparatus, system, module or network for enabling secure data communication with a device, the communication being between a server and a client associated with the device, wherein the security elements are retrieved using a bootstrapping protocol. The device can be an M2M device.

The secure communication may be for the purpose of managing the device and/or the client and/or services (e.g. provided by the device) by the server. Both the device and the server may be machines (i.e. not requiring any human intervention to work). When the device is a machine, the server may be used to manage it. Again, the management may be done without any human intervention (e.g. automatically).

As discussed above, the solution could be used in conjunction with the LWM2M protocol, but the solution could be extended to other Device Management protocols, or to other M2M services (e.g. securing delivery of binary SMS). In particular, and as discussed above, the use of the solution in conjunction with an M2M-specific protocol, such as LWM2M, allows the solution to be very efficient when used with M2M devices, and in particular, when used to manage the device and/or services provided by (or via) the device. In other words, all the advantages mentioned above are further enhanced and optimised when the solution is used in conjunction with an M2M-specific protocol.

In addition, there is also provided any aspects or combination of aspects according to any one of the claims.

Any combination of the features described in connection with any of the aspects is also provided, even if not explicitly disclosed.

With reference to Figure 2, an exemplary architecture (100) is shown that may be implemented, in particular when GBA is used. A device 110 (in the example, an M2M Device and/or a User Equipment) is associated with a card 112 (in the example, a UICC) and a Client 116 (in the example, a Device Management (DM) client. Note that this client

could also be an LWM2M Client, namely a client that can manage the device itself and service/applications provided by the device e.g. asset control). The device 110 is also associated with a device authentication component 114 (in the example, a GAA server). Further, a server 120 is provided (in the example, a DM server), the server associated with a server authentication component 122 (in the example, a Network Application Function (NAF)). Further, an authentication component 130 is provided (in the example, a Bootstrapping Server Function (BSF)) and a register 140 (in the example, an HLR or HSS). Also, four different interfaces are provided for communication between the various components, in particular interface Ua 150 between device 110 and server 120, interface Ub 160 between device 110 and authentication component 130, interface Zn 170 between authentication component 130 and server 120, and interface Zh/Zh' between authentication component 130 and register 140.

In particular, with reference to GBA, document TS 33.220 defines the following components and interfaces, which are shown on Figure 2. NAF, the "Network Application Function", is a server-side component of an application that may be secured using GBA. In a preferred embodiment, the NAF may be a software component within a Device Management (DM) Server.

Some aspects of a BSF, HLR/HSS, UE, Ua, Ub, Zh/Zh' and Zn are provided in "Details of 3GPP standards and technologies used to implement aspects of the method and system" above.

On successful authentication of the device 110, the BSF 130 derives the shared secret Ks_NAF, which is retrieved by the NAF. In a preferred embodiment, the BSF 130 would most likely be on a separate server from the HLR/HSS 140, but within an M2M platform cluster.

The HLR/HSS may be "GBA-aware" (so that it stores details for a GBA user subscription) or may be a legacy component. In a preferred embodiment, the HLR/HSS would be the HLR or HSS of an M2M mobile operator (i.e. one dedicated specifically to serving M2M connections).

The UE 110 is, in the proposed solution, an M2M device.

In a preferred embodiment, the Ua is the interface between a Device Management client 116 and Device Management server 120.

5 In a preferred embodiment, the Ub would be the interface between the "GAA Server" component 114 of the device and the BSF 130.

In a preferred embodiment, the Zn interface is used.

10 In the proposed solution, this interface is between the Device Management Server 120 and the BSF 130. The WS version of the interface would allow placement of a DM Server in multiple locations (not just in the M2M operator/platform cluster), and allow future NAFs in multiple locations.

15 With reference to Figure 3, the procedure for setting up the secure communication in accordance with the present invention is now described, in particular when GBA is used.

At 205, the UE 110 contacts over interface Ua the NAF 122 (in the described embodiment, the Device Management client 116 contacts the Device Management server 122) and discovers that the NAF requires it to acquire a shared secret using GBA. This could be
20 because there is no existing secret, or the existing secret has expired, or is otherwise considered invalid by the NAF.

The exact interface and communication method may be specific to the application concerned. One possible interface and communication method for OMA Lightweight M2M
25 is discussed below.

Over the internal UE interface from DM client to GAA server: at 210, the DM client 116 requests the GAA server 114 to obtain a shared secret. It presents an identifier for the corresponding NAF (NAF_Id).
30

Over the Ub Interface: at 215, The UE 110 contacts the BSF (GAA Server 114 contacts the BSF 130). This may be a basic http GET request. The UE presents an "IMPI" (equivalent of an IMSI) or a "TMPI" (equivalent of a TMSI) for anonymity reasons, if one is available.

35 Over the Zh or Zh' Interface: at 220, the BSF 130 requests an authentication vector from the HLR/HSS 140. At 225, the HLR/HSS 140 returns a fresh vector, consisting of a

RAND, AUTN, XRES, CK, and IK, for example.

5 The BSF 130 generates a transaction identifier (B-TID) and passes (230) the B-TID together with the RAND and AUTN back to the UE 110. It may also indicate the lifetime of the B-TID, and the associated key.

10 Over the internal UE interface from the GAA Server to the UICC: at 235, the GAA Server 114 forwards the RAND and AUTN to the UICC 112 which validates the AUTN. If the AUTN is valid, then the BSF 130 is authenticated. At 240, the UICC 112 returns a RES, CK and IK to the GAA Server 114.

15 At 245, the UE 110 (GAA Server 114) contacts the BSF 130 again, using the resulting RES for HTTP Digest authentication (which is identified in "Details of 3GPP standards and technologies used to implement aspects of the method and system" above).

20 The BSF 130 verifies the HTTP Digest using the XRES. If it matches, then the UE 110 has been successfully authenticated. The BSF 130 stores the tuple <IMPI, B-TID, RAND, CK, IK> and tells at 250 the UE 110 that the authentication was successful. The UE 110 stores <B-TID, RAND, CK, IK>.

25 Over the internal UE 110 interface from DM client 116 to GAA server 114: the UE 110 (GAA Server 114) derives a secret Ks_NAF using the CK, IK, RAND, IMPI and NAF_Id. At 255, it passes Ks_NAF and the B-TID back to the DM client 116.

30 Over the Ua interface again: at 260, the UE 110(DM Client 116) contacts the NAF (DM Server 122) and presents the B-TID as retrieved above.

35 Over the Zn Interface: at 265, the NAF 122 contacts the BSF 130, and presents the BTID. The BSF 130 authenticates the NAF, derives the corresponding Ks_NAF, and at 270 returns it to the NAF, together with an indicator of key lifetime.

The UE 110 (DM Client 116) and NAF (DM Server 122) now both share Ks_NAF. They can use it directly, or to derive their own session keys for further communication.

40 Again, the exact interface and communication method may be specific to the application concerned. One possible interface and communication method for OMA Lightweight

M2M is discussed below.

As discussed above, the solution could be used in conjunction with the LWM2M standard.

This standard can be viewed as a successor to existing OMA Device management

5 standards (OMA DM 1.0 to 1.3), but heavily optimized for low end machine-type devices,

and with an extended management scope beyond the device itself including the

management of services provided by the M2M device such as asset control. This contrasts

for instance with OMA DM 2.0 which is the successor for consumer devices like smart-

phones, tablets etc. Other widely-used Device Management standards include TR-069,

10 which was developed by the Broadband Forum for managing Customer Premises

Equipment (in particular DSL modems).

The exemplary flow described with reference to Fig. 3 is very generic, and can be used

with many different sorts of device management protocols (or other application protocols).

15 As can be seen, many details of the Ua interface are outside the scope of 3GPP (RTM)

and are left to other standards to complete (or left to proprietary implementations).

However, integration with the LWM2M standard is possible, as described in these
examples.

20 Under the specification (see above), the security for OMA LWM2M is based on DTLS v1.2

(see above) and CoAP (see above). Both the client and server must support pre-shared

key DTLS (e.g. see section 7.1.1, page 41), whereas support for certificate-based

authentication is only optional. This means that a key derived by GBA (Ks_NAF) could be

used as a DTLS pre-shared key and it would work with any DM client/DM server pair.

25

The general approach for pre-shared key TLS is referenced in "Details of 3GPP standards

and technologies used to implement aspects of the method and system" above. The GBA

and TLS-PSK protocols work well together. In 205 described above, the "Server Hello"

message contains a field where the server can indicate that it supports GBA-bootstrapping,

30 and in response, the client can then provide an identifier (B-TID) for an already

bootstrapped key (260). Or if the client doesn't already have a bootstrapped key, it asks the

GAA server to get one, before resuming the "Client Hello" and "Server Hello" at 260. The

use of the Ks_NAF to derive session keys is then specified entirely within the TLS-PSK

protocol. The 3GPP (RTM) spec assumes HTTP /TLS, but the basic approach looks the

35 same for CoAP / DTLS.

27 07 20

To improve consistency with the OMA profile of GBA, the LWM2M spec may need to define a “protocol identifier” for DTLS pre-shared key and have it registered by OMNA (see section 5.2.1 of OMA GBA Profile, Approved Version 1.1 – 31 Jul 2012 found at http://technical.openmobilealliance.org/Technical/release_program/sec_cf_archive.aspx).

5

Aside from GBA aspects, the M2M device may be configured to support the security of OMA LWM2M, which is referenced in “Details of 3GPP standards and technologies used to implement aspects of the method and system” above.

10

Additional aspects

1. Device Development for GBA

15 As can be seen from Figure 2 and Figure 3, the M2M device may contain several internal components. It should support a DM client which is “GBA aware”, as well as a “GAA Server” component.

The GAA Server component should support internal interfaces to the DM client and to the
20 SIM card (UICC) as well as the external Ub interface to the BSF. The interface to the UICC may be particularly challenging, as the M2M device may not expose an existing API to allow device software to send commands to the UICC. One possibility (that may be used) is for the modem to expose AT commands. However, this may not be at a sufficiently low level (AT+CSIM allows raw APDUs to be communicated to the UICC) in every case.

25 Further, there may be security issues: while the GAA Server must be able to interface to the UICC, general applications installed on the device should not be able to use this interface, as that could allow external parties to impersonate the device (and engender fraud on the cellular network). So the API to the SIM Card should be privileged, as well as being sufficiently low level to be usable.

30

2. Ub tunnelling, or GBA Push

The interface to the BSF is based on http and HTTP Digest authentication. One alternative may be “tunnelling” the Ub interface within the Ua interface, so that the device only needs
35 to support the CoAP protocol (not HTTP as well).

A related alternative is using the GBA "Push" variant, and carrying push messages (Upa interface) within the Ua interface. Both of these would require identifying suitable commands and parameters in the Ua interface (i.e. the relevant Device Management protocol) to carry the tunnel or push messages. The interfaces and message flow for GBA push are outlined below (see also 3GPP TS 33.223, entitled "3G Security; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) Push function", it can currently be retrieved by <http://www.3gpp.org/ftp/Specs/html-info/33223.htm>).

10 With reference to Figure 4, an example Processing and message flow for GBA Push follows:

1. A NAF establishes a shared NAF SA with a UE which is registered for Push services. It knows the identity of the subscriber.
- 15 2. The Push-NAF generates the GPI (GBA Push info) Request and sends the GPI Request to the BSF.
3. Upon receiving the request from the NAF, the BSF checks that the NAF is authorized, and resolves the requested subscriber identifier to a private identifier (e.g. IMSI).
4. The BSF fetches a new AV (authentication vector) and subscriber's GUSS (GBA User Security Settings) from the HSS.
- 20 5. The HSS sends the AV and the GUSS to the BSF.
6. When the BSF receives the AV Response from the HSS, it generates the NAF keys based on the requested NAF_Id and creates the relevant GPI Response.
7. The BSF sends the GPI Response to the NAF.
- 25 8. The NAF stores the received information together with other user information in a NAF SA.
9. The NAF then forwards the GPI to the UE over Upa using the selected transport mechanism and the given transport address.
10. When the UE receives the message containing the GPI, it processes the GPI as for regular GBA, and stores the corresponding NAF SA(s)
- 30

The UE and NAF are now ready to use the established NAF SA.

TR33.223 specifies that Upa is a new interface that is separate from Ua – "a new reference point Upa is introduced between the NAF and the UE" (Section 4.2.1). As such, the Ua interface should be unaware of whether GBA or GBA-push is being used.

3. Provisioning the address of the BSF and the NAF

The address of the BSF (http URL) may be pre-loaded when the device is manufactured. It could be device managed itself, which would seem to create a “chicken-and-egg” problem, but the DM Server could, for instance, provide an address for an acceptable BSF in the ServerHello. Or http traffic might be routed by the M2M mobile operator to a default BSF address. Similarly, the location of the preferred DM Server might need to be pre-loaded, or the M2M mobile operator could route CoAP traffic to a default DM Server address.

4. Flavour of GBA (GBA-ME, GBA-U, GBA-SIM etc.)

Several different versions of GBA are referenced in “Details of 3GPP standards and technologies used to implement aspects of the method and system”. GBA-U has security advantages, but also logistic advantages: it permits a longer lifetime for the B-TID as the derived key is stored more securely. It allows safe retention of Ks during power-off cycles for instance. GBA-U requires specific support from the UICC, so would have a (modest) increment to the cost. Since M2M devices are typically provided with a new UICC anyway at manufacture, it is a software/development cost rather than a hardware cost. Also, in a model with a customised UICC, this may allow for a solution using restricted AT commands to the modem, rather than full AT+CSIM.

5. Location of the NAF (DM Server) and type of Zn interface

The architecture example allows for there to be several DM Servers in different locations: it could be part of an M2M platform (e.g. M2M mobile operator) cluster, or hosted elsewhere by a network operator/service provider, or perhaps by a customer of said operator/provider. The BSF may need to be located within a firewalled Demilitarized Zone (DMZ), or perhaps connected via an http proxy in the DMZ (so allowing external http Web Service access from NAFs), and then would execute the Diameter interface to the HLR/HSS. It may be undesirable to expose an http interface directly onto the server supporting the HLR, or to tunnel Diameter through firewalls. However, if the DM Server is itself part of the M2M platform cluster then this may be over-engineering. Possibly, a Diameter solution for the Zn interface then becomes acceptable.

6. Use of Zh or Zh' Interface

Ideally, the HLR may be upgraded to a full HSS with support for the Zh reference point. However, if the HLR/HSS only supports Zh' then the BSF will need to be more complicated, and take on some of the subscription management functions (profiling, lifetime, security
5 policies) typically associated with the HSS.

7. Development of NAF component

While the NAF functionality looks fairly straightforward, it will need to be developed for each
10 DM Server used, and for each additional application which uses GBA.

GBA keys could be used to protect SMS (e.g. encrypt/ integrity protect SMS using a secure packet interface e.g. like ETSI TS 102.225 which is used for SIM OTA). This SMS channel is likely to be more efficient than DTLS.

15

In addition, regardless of GBA, a secure SMS protocol could be linked to a Device and/or Service management protocol, namely: using a secure SMS protocol (e.g. originally designed for SIM OTA (102 225)), but now adapted for LWM2M communications, combined with using the LWM2M protocol to define (and manage) the necessary
20 parameters for the secure SMS protocol (i.e. the relevant K_{ic}, K_{ID}, SPI, TAR, and keys).

GBA could be used to securely derive the keys.

Further aspects and advantageous or preferable features are described in the following
25 paragraphs.

LWM2M needs a security solution for the SMS bearer. Without a solution, SMS will not be usable as a bearer, severely limiting scope of LWM2M. A solution to this problem is to use SIM OTA security (e.g. see TS 102 225).

30

TS 102.225 relies on the keys and parameters being already agreed between client and server. However, it is difficult to pre-load these into LWM2M client devices, and ensure that they are sent to servers, because there is no present infrastructure for doing so. It would be pointless to deliver the keys and parameters over unsecured SMS.

35

There are various proposed solutions for delivering these keys and parameters in a secure way.

5 In a first solution, there is provided switching bearer to UDP/Coap and running DTLS. The, DTLS session can be used to secure the LWM2M Bootstrap protocol. The LWM2M Bootstrap can be used to set the TS 102.225 keys and parameters securely. Note that managed resources/objects need to be defined to allow the Bootstrap server to update them; the format of these resources is specified in the numbered paragraphs below.

10 In a second solution, there is provided relying on a SIM card (UICC) which has already having been provisioned with keys and parameters, and using this card to terminate TS 102 225 security. Please note that, because this solution provides a secure channel, the same channel can be used to deliver other keys and parameters.

15 In a third solution, there is provided use of GBA to set up the keys and parameters. This works because the GPI (GBA Push Info) can be delivered over unsecured SMS. So, there is no requirement to have an initial key to protect the SMS. (Note that the delivery of the parameters like Kic, KID, SPI and TAR is not obvious, but these are only 6 bytes, and there are fields in the GPI e.g. App_Lbl, NAF_Id, P-TID which could be used to carry this info.)

20 Further details are provided in the numbered paragraphs below.

UDP channel security for [COAP] is referenced in "Details of 3GPP standards and technologies used to implement aspects of the method and system" above.

25 Since the LWM2M protocol utilizes DTLS for authentication, data integrity and confidentiality purposes, the LWM2M Client and LWM2M Server SHOULD keep a DTLS session in use for as long a period as can be safely achieved without risking compromise to the session keys and counters. If a session persists across sleep cycles, encrypted and
30 integrity-protected storage SHOULD be used for the session keys and counters.

Note that the Client-Server relationship of DTLS (i.e. who initiated the handshake) is separate from the Client-Server relationship of LWM2M.

35 Considering that any device with a LWM2M Client can be managed by any LWM2M Server and LWM2M Bootstrap Server the choice of Cipher Suites is not limited to the list defined in

Section 9 of [CoAP]. Due to the sensitive nature of Bootstrap Information, particular care has to be taken to ensure protection of that data including constraints and dependencies within a LWM2M Client/ Bootstrap Server relationship according to the adopted security mode.

5

Concerning Bootstrap from a Smartcard, the same care has to be taken and a secure channel between the Smartcard and the LWM2M Device SHOULD be established as described in Appendix H of OMA LWM2M in reference to GlobalPlatform Secure Channel Protocol 03 (SCP 03) Amendment D v1.1 Sept 2009.

10

The keying material used to secure the exchange of information using a DTLS session may be obtained using one of the bootstrap modes referenced in "Details of 3GPP standards and technologies used to implement aspects of the method and system" above.

15

The Resources (i.e. "Security Mode", "Public Key or Identity", "Server Public Key or Identity" and "Secret Key") in the LWM2M Security Object that are associated with the keying material are used either

1) for providing UDP channel security in "Device Registration", "Device Management & Service Enablement", and "Information Reporting" Interfaces if the LWM2M Security Object

20

Instance relates to a LWM2M Server, or,

2) for providing channel security in the Bootstrap Interface if the LWM2M Security Object instance relates to a LWM2M Bootstrap Server.

25

LWM2M Clients MUST either be directly provisioned for use with a target LWM2M Server (Manufacturer Pre-configuration bootstrap mode) or else be provisioned for secure bootstrapping with an LWM2M Bootstrap Server. Any LWM2M Client which supports Client or Server initiated bootstrap mode MUST support at least one of the following secure methods:

30

1) Bootstrapping with a strong (high-entropy) pre-shared secret, as described in section 7.1 of OMA LWM2M. The cipher-suites defined in this section MUST NOT be used with only a low-entropy pre-shared secret.

2) Bootstrapping with a temporary, low-entropy pre-shared secret (such as a PIN, password and private serial number) using the cipher-suite

TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256, as defined in RFC5489.

3) Bootstrapping with a public key or certificate-based method (as described in sections 7.1.2 and 7.1.3 of OMA LWM2M). The LWM2M client MUST use a unique key-pair, one which is unique to each LWM2M client.

- 5 For full interoperability, a LWM2M Bootstrap Server SHALL support all of these methods.

NOTE: The above security methods can also be used by the LWM2M Bootstrap Server to provision Klc and KID for SMS channel security (see below for SMS channel security).

10 **SMS Channel Security**

The SMS channel security is provided by the Secured Packet Structure [ETSI TS 102 225] and [SCP080] which is defining secured packets for different transport mechanisms.

- 15 The solution was originally designed for securing packet structures for UICC based applications, however, for LWM2M it is suitable for securing the SMS channel between client and server.

- 20 The SMS channel security specified in this section MUST be applied when the SMS binding is used.

- When the LWM2M device supports a smartcard, the security SHOULD be terminated on the smartcard. The LWM2M client SHOULD pass SMS messages to the smartcard for encryption and integrity protection before sending, and SHOULD pass encrypted SMS
25 messages received from the LWM2M server to the smartcard for decryption and integrity checking.

- A LWM2M Client which supports the SMS binding SHALL support the Secured Packet Structure as defined in [ETSI TS 102 225] and [SCP080]. The LWM2M Client SHALL share
30 the relevant keys - identified by Klc and KID - with a LWM2M Bootstrap Server during bootstrapping, or with a LWM2M Server otherwise.

- A LWM2M Bootstrap Server which supports the SMS binding SHALL support the Secured Packet Structure as defined in [ETSI TS 102 225] and [SCP080].
35

A LWM2M Server which supports the SMS binding SHALL support Secured Packet Structure as defined in [ETSI TS 102 225] and [SCP080].

The following applies to LWM2M Client and LWM2M Bootstrap Server and LWM2M
5 Server:

- Single DES SHALL NOT be relied on.
- AES or Triple DES with three different keys MUST be used.
- Preferably, AES should be used. Where AES is used it should be used with
10 CBC mode for encryption (see coding of Klc in [ETSI TS 102 225] section 5.1.2) and in CMAC mode for integrity (see coding of KID in [ETSI TS 102 225] section 5.1.3).
- SPI SHALL be set as follows (see coding of SPI in [ETSI TS 102 225] section 5.1.1).:
15
 - cryptographic checksum
 - ciphering
 - process if and only if counter value is higher than the value in the RE
- Preferably, TAR (see coding of TAR in [ETSI TS 101 220], section 6) SHALL be set to a value in the range BF FF 00 - BF FF FF.

20
NOTE: A TAR for LWM2M SMS security will be requested from ETSI SCP and the range above applies only until the TAR has been assigned.

Preferably, Klc, KID, SPI and TAR SHALL be stored in the “SMS Binding Key Parameters”
25 Resource.

Preferably, the corresponding key values should be stored in the “SMS Binding Secret Keys” Resource.

30 A LWM2M Client which uses the SMS binding may either be directly provisioned for use with a target LWM2M Server (Manufacturer Pre-configuration bootstrap mode) or else be able to bootstrap via the UDP binding.

A LWM2M Client, Server or Bootstrap Server supporting SMS binding SHALL discard SMS
35 messages which are not correctly protected using the expected parameters stored in the

“SMS Binding Key Parameters” Resource and the expected keys stored in the “SMS Binding Secret Keys” Resource, and SHALL NOT respond with an error message secured using the correct parameters and keys.

5 **LWM2M Object: LWM2M Security**

Description: This LWM2M object provides the keying material of a LWM2M Client appropriate to access a specified LWM2M Server. One Object Instance SHOULD address a LWM2M Bootstrap Server

10

These LWM2M object resources MUST only be changed by a LWM2M Bootstrap Server or SmartCard provisioning and MUST NOT be accessible by any other LWM2M Server.

Example Object Info:

15

Object	Object ID	Object URN	Multiple Instances?	Mandatory?
LWM2M Security	0		Yes	Yes

Resource Info:

Resource Name	Type	Range or Enumeration	Units	Descriptions
LWM2M Server URI	String	0 – 255 bytes	-	Uniquely identifies the LWM2M Server or LWM2M Bootstrap Server, and is in the form: “coaps://host:port”, where host is an IP address or FQDN, and port is the UDP port of the Server.
Bootstrap Server	Boolean		-	Determines if the current instance concerns a LWM2M Bootstrap Server (true) or a standard LWM2M Server

				(false)
Security Mode	Integer	0-3	-	Determines which UDP channel security mode is used 0: Pre-Shared Key mode 1: Raw Public Key mode 2: Certificate mode 3: NoSec mode
Public Key or Identity	Opaque		-	Stores the LWM2M Client's Certificate (Certificate mode), public key (RPK mode) or PSK Identity (PSK mode). The format is defined in Section E.1.1.
Server Public Key or Identity	Opaque		-	Stores the LWM2M Server's or LWM2M Bootstrap Server's Certificate (Certificate mode), public key (RPK mode) or PSK Identity (PSK mode). The format is defined in Section E.1.1.
Secret Key	Opaque		-	Stores the secret key or private key of the security mode. The format of the keying material is defined by the security mode in Section E.1.1. This resource MUST only be changed by a bootstrap server and MUST NOT be readable by any server.
SMS Security Mode	Integer	0-255		Determines which SMS payload security mode is used (see section 7.2) 0:Reserved for future use

				<p>1:Secure Packet Structure mode device terminated</p> <p>2: Secure Packet Structure mode smartcard terminated</p> <p>3: NoSec mode</p> <p>255: Proprietary modes</p>
SMS Binding Key Parameters	Opaque	6 bytes	-	Stores the Klc, KID, SPI and TAR. The format is defined in Section D.1.2
SMS Binding Secret Keys	Opaque	32-48 bytes	-	Stores the values of the keys for the SMS binding. This resource MUST only be changed by a bootstrap server and MUST NOT be readable by any server.
LWM2M Server SMS Number	Integer			MSISDN used by the LWM2M Client to send messages to the LWM2M Server via the SMS binding. The LWM2M Client SHALL silently ignore any SMS not originated from unknown MSISDN
Short Server ID	Integer	1-65535	-	<p>This identifier uniquely identifies each LWM2M Server configured for the LWM2M Client.</p> <p>This resource MUST be set when the Bootstrap Server resource has false value.</p> <p>Default Short Server ID (i.e. 0) MUST NOT be used for identifying the LWM2M Server.</p>

Client Hold Off Time	Integer		s	Relevant information for a Bootstrap Server only. The number of seconds to wait before initiating a Client Initiated Bootstrap once the LWM2M Client has determined it should initiate this bootstrap mode
-----------------------------	---------	--	---	---

UDP Channel Security: Security Key Resource Format

5 This section defines the format of the Secret Key and Public Key and Identity resources of the LWM2M Server and LWM2M Bootstrap Objects when using UDP Channel security. These resources are used to configure the security mode and keying material that a Client uses with a particular Server. The Objects are configured on the Client using one of the Bootstrap mechanisms described in Section 5.1 of OMA LWM2M. The use of this keying material for each security mode is defined in Section 7.1 of OMA LWM2M.

10

Pre-Shared Key (PSK) Mode

15 The PSK is a binary shared secret key between the Client and Server of the appropriate length for the Cipher Suite used [RFC4279]. This key is composed of a sequence of binary bytes in the Secret Key resource. The default PSK Cipher Suites defined in this specification use a 128-bit AES key. Thus this key would be represented in 16 bytes in the Secret Key Resource.

20 The corresponding PSK Identity for this PSK is stored in the Public Key or Identity resource. The PSK Identity is simply stored as a UTF-8 String as per [RFC4279]. Clients and Servers MUST support a PSK Identity of at least 128 bytes in length as required by [RFC4279].

Raw-Public Key (RPK) Mode

25

The raw-public key mode requires a public key and a private key of the appropriate type and length for the Cipher Suite used. These keys are carried as a sequence of binary bytes with the public key stored in the Public Key or Identity Resource, and the private key stored

in the Secret Key Resource. The default RPK Cipher Suites defines in this specification use a 256-bit ECC key. Thus the Certificate Resource would contain a 32 byte public key and the Secret Key Resource a 32 byte private key.

5 Certificate Mode

The Certificate mode requires an X.509v3 Certificate along with a matching private key. The private key is stored in the Secret Key Resource as in RPK mode. The Certificate is simply represented as binary X.509v3 in the value of the Public Key or Identity Resource.

10

SMS Payload Security: Security Key Resource Format

This section defines the format of the Secret Key and Public Key and Identity resources of the LWM2M Server and LWM2M Bootstrap Objects when using SMS Payload security.

- 15 These resources are used to configure keying material that a Client uses with a particular Server. The Objects are configured on the Client using one of the Bootstrap mechanisms described in Section 5.1. The use of this keying material is defined in Section 7.2. The SMS key parameters are stored in the order Klc, KID, SPI, TAR (Klc is byte 0).

- 20 Ordering of bits within bytes SHALL follow ETSI TS 102 221 "Coding Conventions" (b8 MSB, b1 LSB).

Unbootstrapping

- 25 If a Security Object Instance is to be deleted, some related resources and configurations need to be deleted or modified. Therefore when Delete operation is sent via Bootstrap Interface, the Client MUST proceed following procedure.

1. If there is an Object Instance that can be accessed only by a Server of the Server Object Instance (i.e. the Server is Access Control Owner and the LWM2M Server can access the Object Instance only in an Access Control Object Instance), the Object Instance and the corresponding the Access Control Object Instance MUST be deleted
 2. If an Object Instance can be accessed by multiple Servers including the Server, which Security Object Instance is to be deleted, then:
 - An ACL Resource Instance for the Server in Access Control Object Instance for the
- 35 Object Instance MUST be deleted

- If the Server is Access Control Owner of the Access Control Object Instance, then the Access Control Owner MUST be changed to another Server according to the rules below: The Client MUST choose the Server who has highest sum of each number assigned to an access right (Write: 1, Delete: 1) for the Access Control Owner. If two or more Servers
- 5 have the same sum, the Client MUST choose one of them as the Access Control Owner.
 3. Observation from the Server MUST be deleted
 4. The Server Object Instance MUST be deleted
 5. Client MAY send “De-register” operation to the Server

- 10 Note: To monitor the change of Access Control Owner, the Server MAY observe Access Control Owner Resource.

As a further enhancement, the GBA BSF may be merged into the LWM2M bootstrap server. This makes it easier to discover a single server. This is shown in Figure 5 in which

- 15 a combined server 420 holds both the BSF 130 and the LWM2M 440.

Extensions for use with Public Key solutions

One problem with the key(s) derived using GBA is that they are based on the long-term secrecy of the underlying (U)SIM key, K or Ki. If that key ever leaks in the future, or if the operator is forced to disclose it for some reason, then any derived keys ever used in GBA may be considered compromised. An adversary or third party who had been recording all conversations with a device may potentially recover all past keys and messages protected by them.

One solution for this is to ensure that the GBA keys are only used temporarily when enrolling in a public key scheme: that scheme may for instance support “perfect forward secrecy” which avoids the problem of future key compromise leading to a recovery of all encrypted messages.

The 3GPP (RTM) standard (GAA) provides some methods for using GBA to enrol for public key certificates. However, but they are particularly complex, and the necessary public key infrastructure (PKI) may be too great an overhead for LWM2M or other M2M protocols.

Authenticate “Raw” public keys from the client:

The M2M device may have a preloaded public-private keypair, or may generate a public-private key-pair on first power up. It presents its "raw public key" (or a hash of the raw public key) to the BSF, protected using HTTP digest and RES (this could be over a tunnelled Ub protocol, since that already requires some customisation). Alternatively, the raw public key may be presented to the NAF (LWM2M server or LWM2M bootstrapping server) protected using the GBA-derived key Ks_NAF. (The device could, for example, present its raw public key over CoAP, integrity protected using Ks_NAF, or present it within a CoAPs session secured using a pre-shared-key cipher suite and Ks_NAF).

10 The DM server (or bootstrapping server) may then use one of the public key ciphersuites, such as:

- TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 as defined in Section 9.1.3.2 of [CoAP]
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in [RFC5289]

15

Alternatively, the DM Server may use a mixed pre-shared key/public key ciphersuite like:

- TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256, as defined in [RFC5489] using the Ks_NAF as the pre-shared key.

20 Preferably, these ciphersuites may provide perfect forward secrecy. There may be several procedures that can be carried out before a Ks_NAF is established. In particular, these procedures may be carried out for devices that don't support GBA, or where the SIM card of the M2M device is not yet provisioned for GBA.

25 The device may present its raw public key unauthenticated, and the DM server or Bootstrapping may accept this raw public key temporarily. The device may use a weak secret (like PIN+IMSI+IMEI) together with TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256, for example, to secure a first connection with a bootstrap server.

30

These back-up methods provide no (or weak) authentication of the device public key, but they at least allow the protocol to get started, including providing the connectivity to tunnel GBA or run GBA-push, and so establish a subsequent strong authentication of the device public key.

35

Authenticate "Raw" public keys from the server.

As an alternative, or in addition to the above-mentioned procedure for authenticating "Raw" public keys from the client, the DM server may present its own "raw public key" to the client, protected by HTTP digest and RES, or other techniques, or by an integrity check or PSK session set up using the GBA-derived key Ks_NAF, for example. The similar cipher-
5 suites to those described above may also be used to provide perfect forward secrecy.

If there has been no opportunity to establish a Ks_NAF then the device may require a preloaded copy or digest of the expected public key (e.g. the public key used by the LWM2M Bootstrap server) or a pre-loaded weak secret. This digest or secret may be
10 retrieved from the SIM card (UICC) to avoid the need to provision directly to the device at manufacture. In an alternative, it may be sent to the device by SMS or the device may accept the raw server public key unauthenticated, at least at first connection.

Again, these back-up methods provide no (or weak) authentication of the server public key,
15 but they at least allow the protocol to get started, including providing the connectivity to tunnel GBA or run GBA-push, and so establish a subsequent strong authentication of the server public key.

The above-mentioned techniques for authenticating "Raw" public keys from the client may
20 also apply to certificate modes e.g. instead of a raw public key, the device may present a self-signed certificate (or certificate signed by an unknown CA). The server may present a self-signed certificate, or an unknown root CA certificate to M2M device. This also avoids the complexity and expense of provisioning the M2M device at manufacture with a public key certificate signed by a pre-existing CA (e.g. device manufacturer's CA) or provisioning
25 multiple root certificates.

These embodiments also allow the frequent refresh of private/public keys or certificates, which addresses some revocation problems. However, there is no need to do revocation checking, as the private/public key doesn't last long. The lifetime of the private/public key
30 may be linked to the lifetime of a B-TID (or key_LT).

Alternatively, GBA could be used to secure messages from a trusted timesource (e.g. a network time), or provide a secure channel to retrieve certificate validation information (CRLs or OCSP responses.) This can address some problems with obtaining correct and
35 up-to-date information for certificate validation.

Figure 8 shows a flowchart of a method 600 for securely communicating with an M2M device in a way to protect the long term secrecy of communications. At step 610 a shared secret is obtained. This may be generated or retrieved from storage. Furthermore, shared secret may be obtained from the M2M device, the server or another source. The shared secret (or data derived from the shared secret, such as its hash) are shared between the M2M device and the server. This may be by simple transmission, SMS or other mechanism, for example. Authentication by the M2M device and/or server may also take place and may use the shared secret.

10 A connection is established, secured or initiated between the M2M device and the server at step 630. This connection utilises the shared secret or derived data. There may be some, weak (e.g. password, PIN, IMSI and/or IMEA derived secret) or no security or authentication involved in the connection. However, the shared secret may now be used to establish cryptographic material between the M2M device and the server at step 640. In other words, the connection may have little or no security and the shared secret may have a low or lower level of security (as compared with the cryptographic material) but may nevertheless be used to provide corresponding cryptographic material at the M2M device and server. The cryptographic material may be used to secure communication between the M2M device and the server (step 650). This secured communication uses a cryptographic protocol (e.g. TLS or SSL) and preferably a cryptographic protocol that provides perfect forward secrecy. The M2M device and the server may now communicate securely.

25 The shared secret may be a public key of a private key and public key pair, or a certificate (including self-signed, signed by a known certificate authority or signed by an unknown certificate authority), for example.

30 Preferably, the shared secret (e.g. key or certificate) is set to expire or be refreshed at intervals (e.g. every one, two, three or more days). The cryptographic material may be of a similar or different type to the shared secret. It may be more secure and may also have an expiry time (although this may be different to that of the shared secret).

35 Figure 9 illustrates schematically a system 700 used to implement the method 600 described with reference to figure 8. The system includes a server 720 including logic 730 to carry out the method. One or more M2M devices 110 communicate over connections with the server 720 in the manner described above. The M2M devices also include logic

710 used to implement the method. In the embodiment shown in figure 9, each M2M device 110 has a single connection with the server 720. In this implementation, the secure communication can be secured over this (typically) unsecure connection. However, in other embodiments a separate connection (of the same or of a different type) may be used
5 for the secure communication to take place.

In the embodiments where the server 720 generates or retrieves the shared secret (e.g. key or certificate) then its own logic 730 will carry out this task. In the embodiments where the M2M Device 110 generates or retrieves the shared secret (e.g. key or certificate) then
10 its own logic 710 will carry out this task.

The following describes particular uses of the perfect forward security and authentication of raw public key procedures or other secure communication methods. These include secure firmware updates, and secure broadcasts.
15

A M2M client device may be advised of a different server-side public key. This different public key may be used to sign firmware updates (over a unicast, broadcast or multicast or peer-to-peer system, for example).

20 This functionality may be extended to other sorts of secure broadcast, for example distributing a public key to sign public warning messages (bomb alerts, tsunami alerts), emergency messages (ambulance approaching, get out of the way), traffic alerts etc.

This procedure has the advantages of reduced or no PKI, and no or easier certificate validation. A public key (protected by a key established using GBA) may be sent where the
25 public key is then used to verify broadcast messages or firmware updates, for example.

The LWM2M specification doesn't have a broadcast channel and firmware updates in LWM2M operate as follows:
30

- The LWM2M Server performs a "Write" (or "Create") operation to alter (or create a new instance of)
a "Firmware Update" object. It may write the update itself to a "Package" resource, but since updates are often large (and difficult to carry over CoAP), it most likely writes a URI
35 to the "Package URI" resource.

- If the "Package URI" resource has been written to, then the LWM2M Client "performs the download at the next practical opportunity". While downloading, the LWM2M client sets the "State" resource to "2" (which means "Downloading"). On completion of download, the LWM2M client sets the "State" resource to "3" (which means "Downloaded"). If the "Package" resource was written to directly, then when the LWM2M client has completed the Write operation, it sets the "State" resource to "3" straight away.
- The LWM2M server can perform a "Read" operation whenever it likes in order to find out the value of the "State". When the state is "3" it can then perform an "Execute" operation on the "Update" resource, which instructs the LWM2M client to install the firmware update.
- The LWM2M client sets the "Update Result" resource to indicate whether the update succeeded or failed, and any reason for failure. The client also sets the "State" resource back to "1" (which means "Idle").
- The LWM2M server can perform another "Read" operation whenever it likes in order to find out the value of the "Update Result" and "State" resources.

No verification of the update package occurs in this process.

However, the following describes improvements to this method. These are illustrated in two example mechanisms. In these examples, a key is carried within LWM2M operations.

Mechanism 1 The LWM2M server performs a "Write" operation on the "Package" resource, but rather than writing the firmware update itself, it writes the (typically public) key that the LWM2M client requires in order to verify the firmware update. The LWM2M server simultaneously (or sequentially) performs a "Write" operation on the "Package URI" resource, telling the LWM2M client where it can find the update itself.

NOTE: The URI here may well take the form of a broadcast URI, for example - see

<http://tools.ietf.org/html/rfc2838> - or

http://www.etsi.org/deliver/etsi_ts/102800_102899/102851/01.01.01_60/ts_102851v010101.pdf

Once the LWM2M client has downloaded the update, then it can verify it using the public key that was written in the “Package” resource.

- Mechanism 2** The LWM2M server performs a “Write” operation on the “Package URI” resource, but the URI contains an encoding of the public key that will be needed to verify the firmware update. For example, take the form of:

dvb://channel135/key=9ICYfQ4RHZO3a3jb6hrQWRQiga7ImCqCaG9JemD7F6r

- Once the LWM2M client has downloaded the update, then it can verify it using the public key that was contained within the “Package URI” resource.

- A key established by GBA may be used to secure a session between the LWM2M client and LWM2M server (for example the Ks_NAF is used as a DTLS pre-shared key) and then the second key is provided within that secured session.

- In one example, residential gas meters may require firmware updates (e.g. a bug is discovered which means that they will incorrectly report gas consumed. Hackers may be trying to exploit this and a patch is required). The cellular operator provides some broadcast capabilities on their network – for example via MBMS. One of the broadcast channels is “free to air” (i.e. it is unencrypted) but consists of a stream of firmware updates for many types of M2M devices (gas meters, electricity meters, engine management systems etc.) Each update may be signed by a separate private key, and the key-pairs may be refreshed regularly (to ensure that only the most recent updates are signed by the latest keys).

- When each gas meter next contacts the device management server, it is provided with an authenticated public key. The public key may be integrity protected using the shared secret (Ks_NAF) that has been established between the gas meter and M2M device management server using GBA. The gas meter then monitors the MBMS broadcast channel to detect when an update is being broadcast and which has been signed with the corresponding private key. It downloads the update, verifies it using the public key, and if verified installs the update.

- In another example implementation, a third party organisation (commercial, government, or non-commercial, for example) may wish to introduce a new service or services and these

services may be categorised by device type and device category (i.e. road traffic alerts may be an information-alert type having a device category as transportation. A tsunami warning system may be of type emergency public safety message, and device category "all". This taxonomical device categorisation system may be further refined to ensure that service advertisements were routed to appropriate devices.)

The third party organisation may inform a service provider (for example, a telecommunications service provider) of the service. The service provider may validate the service and may use the mechanism described above in relation to secure message broadcast to distribute a description of the service to devices.

This service could be used recursively to advise of other service advertisers. In other words, the service provide could advertise the existence of a Regulatory body (A government agency) that could be authorised to advertise government services.

These services may be of three types opt in, opt out and mandatory. In other words, a service for national security purposes advertised by a government agency may be mandatory, a service for road traffic alerts provided by a commercial company may be opt in and a service advertised by a government agency for a public safety purpose may be opt out.

In a further example, a third party organisation (commercial, government, or non-commercial) may wish to introduce one or more new services. These services may be categorised by device type and device category (i.e. road traffic alerts would be of type information-alert, and device category transportation, tsunami warning system would be of type emergency public safety message, and device category all. This taxonomical device categorisation system could be further refined to ensure that service advertisements were routed to appropriate devices.)

The third party organisation may inform a telecommunications network operator of the service. The network operator may validate the service and may use the method of communicating with M2M devices described above to distribute a description of the service to devices.

This service could be used recursively to advise of other service advertisers. In other words, a network operator could advertise the existence of a regulatory body (i.e. a

government agency) that could be authorised to advertise government services, for example.

5 Examples services may be of three types in general: opt in / opt out / mandatory. For example, a service for national security purpose advertised by a government agency would be mandatory, a service for road traffic alerts provided by a commercial company would be opt in and a service advertised by a government agency for a public safety purpose could be opt out.

10 One example use of GBA keys is to protect SMS (encrypt / integrity protect SMS using secure packet interface e.g. like ETSI TS 102.225 which is used for SIM OTA). This SMS channel is likely to be more efficient than DTLS.

15 Encrypted SMS functionality is currently missing from OMA LWM2M. In other implementations, a secured SMS protocol may be used with a Device management protocol. For example:

- 20 a) Using a secure SMS protocol, originally designed for SIM OTA (102 225), but now adapted for LWM2M communications, and
- b) Using the LWM2M protocol to define (and manage) the necessary parameters for a secure SMS protocol, e.g. the relevant K_{ic}, K_{ID}, SPI, TAR, and keys.

25 Use of GBA then becomes a further extension to this idea, as it is used to securely derive the keys.

30 In one particular example, an industrial thermostat system (e.g. in a warehouse) may have the temperature set remotely via a device management server. Since the temperature data is very short (just a number in degrees C), then it (or any other data) may be transmitted in a single SMS, for example using the OMA LWM2M protocol with a SMS bearer.

35 Therefore, there is a need to protect the SMS to prevent malicious tampering with the thermostat (or other equipment that may rely on sensor or parameter data) and possible damage to goods stored in the warehouse, unnecessary energy bills etc. The SMS may be encrypted and integrity protected using the secure packet protocol described in ETSI TS 102 225. First, GBA is used to establish a KS_NAF between the thermostat and the

LWM2M server. For convenience, an initial SMS may be used to deliver a GPI (GBA push info) message from the LWM2M server to the thermostat.

5 This GPI contains an authentication challenge to pass to a SIM card (UICC), which might be in the thermostat, or perhaps elsewhere in the warehouse network (it might be in a communications hub). The challenge is processed by the USIM application on the SIM card, and a CK and IK are returned.

10 These are then used to derive the Ks, and then the Ks_NAF which is passed to the thermostat. The thermostat (or other equipment) then extracts keys for the Kc and Kid and stores them for use by the LWM2M client on the thermostat. A solution could be as follows: The Ks_NAF is 256 bits (or total number of bits); the first 128 bits (or set of bits) are used as an AES encryption key (for the Kc) and the last 128 bits (or next or final set of bits) are used as an integrity key (for the Kid). Once this has been set up, an encrypted and
15 integrity protected SMS with the desired temperature can be sent whenever needed from the LWM2M server to the LWM2M client on the thermostat. The client then decrypts and verifies the desired temperature, and the thermostat sets that temperature (or device changes a particular setting or parameter).

20 Automated test-suite.

In this example implementation, an automated test-suite may be used by added value service providers that could be used to supply additional maintenance services (in this case diagnostics and performance analysis) for the purpose of ensuring optimal device
25 behaviour. Typically, this would be a service offered by the original device vendor as an added-value service. However, it is feasible that specialist service providers might supply these services (e.g. car maintenance/tuning service). The benefit to the equipment vendor is that it creates a revenue and on-sell opportunities. Another use case would be where there was a complex device ecosystem (e.g. a home area network (HAN) with many
30 peripheral devices of different types). A HAN maintenance service provider would have (or would create) signature data for a correct and incorrect operation of supported devices and would provide diagnostic services for those devices.

35 Other messaging channels (and this applies to all the described examples) could be conventional DSL and WiFi (RTM) where there was a low level of security over the physical infrastructure and easy to intercept and interfere with communications.

27 07 20

Use GBA to send a secured message which starts a self-test/diagnostic test of device. This covers LWM2M functionality, or general functionality. Results of the self-test can also be reported back using a GBA-secured message. This could build on SMS security, as described in Idea 16, or use a different messaging channel.

For performance, and power-management reasons, we don't want just anyone to be able to trigger a self-test, which is why this should be a secured message. Also, for privacy and security reasons, we don't want the results of the self-test to be revealed to anyone who asks. However, it is unlikely that anyone would ever set up special strong security just to do this. So it is a good use-case for GBA as opposed to other security architectures.

End-of-life sale and decommissioning of device.

This example implementation illustrates the ability to create a trust relationship between devices and service providers, creates opportunities to offer security services to all devices in a symmetric fashion. For example, a toaster can be used as an appliance to facilitate a user to vote in a Government election, and an e-purse can acquire road traffic information. Security services can be offered on an any-to-any basis in a device agnostic fashion. In the end of life cases, the authorisation for the device to participate in an auction to sell itself or/and to procure a replacement would be supplied by the owner of the appliance (by some means), and the responsibility for participating in an auction sale/bid would be delegated to the device in question.

Once the trust relationship is in place, it creates an opportunity to add value. In the end of life case, this could include providing service/life histories to potential buyers or scheduling collection of the device for disposal and recycling.

The device may use a secure association established using GBA to list itself on eBay for example, and order a replacement.

These sorts of automated sale and order applications require strong security. Therefore, this is a good use of the particular described GBA implementation.

Security Keyring

unsecured device. Many examples can be given for this use case using the keyring analogy. These transfers may be automated or under user control or a combination of both.

- 5 A further extension of this would be symmetric use where two unrelated GBA aware appliances needed to exchange information (as described in WO2012/035340).

27 07 20
10 The devices recognise the presence and the identity of each other (i.e. they have visibility of each other such as a car passing through a toll booth. The toll booth could use an induction loop to detect the presence of a car, and use Bluetooth (RTM)/RFID (or other sensor) to supply other information in plain text, for example) and would acquire signature information from the other device. Each device communicates by secure means to their Device Management platform the existence of the other device + security credentials + the security signatures (received and given). The Device Management platforms (which would
15 have a pre-existing trust relationship between each other, such as by means of trusted CA certificates) then validate the supplied signatures for each device and negotiate security keys to support inter communication between the devices. (This may be a simple symmetric key). The derived keys are sent to the devices and the devices are then able to communicate with each other securely and initiate a monetary transfer (for example).

20 It is to be understood that the above description is given by way of example and only for the benefit of understanding the solution, and it must include also any combination of the above features, as well as any alterations, modifications or otherwise addition which could be done by a skilled person by use of his/her skills and/or general knowledge in the
25 relevant and/or neighbouring art.

Many combinations, modifications, or alterations to the features of the above embodiments will be readily apparent to the skilled person and are intended to form part of the invention. Any of the features described specifically relating to one embodiment or example may be
30 used in any other embodiment by making the appropriate changes.

CLAIMS:

1. A method for securely communicating with a machine to machine, M2M, device comprising the steps of:

5 sharing a secret or data derived from the secret between the M2M device and a server;

establishing a connection between the M2M device and the server;

using the shared secret or data derived from the shared secret to establish cryptographic material on both the M2M device and the server; and

10 securing communication between the M2M device and the server with a cryptographic protocol using the established cryptographic material, wherein the cryptographic material is unrecoverable from the shared secret or data derived from the shared secret alone, wherein the step of using the shared secret or data derived from the shared secret to establish the cryptographic material on both the M2M

15 device and the server further comprises the steps of:

the M2M device generating or obtaining a public key and a corresponding private key;

the server generating or obtaining a public key and a corresponding private key;

20 the M2M device authenticating the public key of the server using the shared secret or data derived from the shared secret;

the server authenticating the public key of the M2M device using the shared secret or data derived from the shared secret;

the M2M device deriving the cryptographic material from the public key of the server and the private key of the M2M device; and

25 the server deriving the cryptographic material from the public key of the M2M device and the private key of the server.

2. The method of claim 1, wherein the cryptographic material is session keys to provide perfect forward secrecy.

30

3. The method of claim 1 or claim 2, wherein the cryptographic protocol is transport layer security, TLS, secure sockets layer, SSL or datagram transport layer security, DTLS.

4. The method according to any previous claim, wherein the shared secret is generated by or stored within the M2M device and the shared secret or data derived from the shared secret are sent to the server.

35

29 04 20

5. The method according to any of claims 1 to 4, wherein the shared secret is generated by or stored within the server and the shared secret or data derived from the shared secret are sent to the M2M device.

5

6. The method according to any previous claim, wherein the data derived from the shared secret is obtained by applying a hash function or other fixed key derivation function to the shared secret.

10 7. The method according to any previous claim, wherein the server is a device management server, device management bootstrap server, a lightweight M2M, LWM2M, server, a LWM2M bootstrap server or a Network Application Function, NAF.

15 8. The method according to any of claims 1 to 7, wherein the server is a Bootstrapping Server Function, BSF.

9. The method according to any previous claim, wherein the connection established between the M2M device and the server is unsecured.

20 10. The method according to any previous claim, wherein the secured communication between the M2M device and the server is over the connection.

25 11. The method of claim 10, wherein the step of securing the connection between the M2M device and the server further comprises providing Generic Bootstrapping Architecture, GBA, or GBA Push Info, GPI, messages over the connection.

12. The method according to any previous claim, wherein the shared secret is used to establish, protect, endorse or authenticate a certificate, a self-signed certificate, or a certificate signed by an unknown or unverifiable certificate authority.

30

13. The method according to any previous claim, wherein the shared secret is used to establish, protect, endorse or authenticate an authority key or certificate, and wherein the authority key is used to sign, endorse or authenticate a public key or certificate for the device and/or server.

35

29 04 20

14. The method according to any previous claim, wherein the shared secret is used to verify that a public key or certificate is still valid.

15. The method according to any previous claim, wherein the shared secret is used to
5 authenticate an updated public key or certificate.

16. The method according to any previous claim, wherein the shared secret is configured to expire at a predetermined time.

10 17. The method of claim 12, wherein the server generates a transaction identifier configured to expire at a predetermined time and further wherein the expiry time of the shared secret corresponds with the expiry time of the transaction identifier.

15 18. The method of claim 15 or claim 16 further comprising the step of obtaining information indicating current time from a trusted time source.

19. The method of claim 18, wherein the information indicating the current time is obtained from messages from the trusted time source secured using GBA.

20 20. A system comprising:
a server;
one or more machine to machine, M2M, devices; and
logic configured to:
share a secret or data derived from the secret between the one or more
25 M2M devices and the server;
establish a connection between the one or more M2M devices and the
server;
use the shared secret or data derived from the shared secret to establish
cryptographic material on both the M2M device and the server; and
30 secure communication between the M2M device and the server with a
cryptographic protocol using the established cryptographic material, wherein the
cryptographic material is unrecoverable from the shared secret or data derived from
the shared secret alone, wherein:
the one or more M2M devices are configured to generate or obtain a
35 public key and a corresponding private key, and wherein the server is further

29 04 20

configured to generate or obtain a public key and a corresponding private key;

the one or more M2M devices are configured to authenticate the public key of the server using the shared secret or data derived from the shared secret;

5 the server is configured to authenticate the public key of the one or more M2M devices using the shared secret or data derived from the shared secret;

the one or more M2M devices are configured to derive the cryptographic material from the public key of the server and the private key of the M2M device; and

10 the server is configured to derive the cryptographic material from the public key of the M2M device and the private key of the server.

21. A computer program comprising program instructions that, when executed on a computer cause the computer to perform the method of any of claims 1 to 19.

15

22. A computer-readable medium carrying a computer program according to claim 21.

23. A computer programmed to perform the method of any of claims 1 to 19.

29 04 20



(51) International Patent Classification:

H04W 4/70 (2018.01) H04L 9/28 (2006.01)

Published:

— with international search report (Art. 21(3))

(21) International Application Number:

PCT/US2019/051481

(22) International Filing Date:

17 September 2019 (17.09.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/732,930 18 September 2018 (18.09.2018) US

(71) Applicant: **IOT AND M2M TECHNOLOGIES, LLC**
[US/US]; 807 Davis Street, Unit 2207, Evanston, IL
60201-7104 (US).

(72) Inventor: **NIX, John, A.**; C/o IoT and M2M Technolo-
gies, LLC, 807 Davis Street, Unit 2207, Evanston, IL
60201-7104 (US).

(74) Agent: **THOMSON, Kirsten, L.**; McDonnell Boehnen
Hulbert & Berghoff LLP, 300 South Wacker Drive, Chica-
go, IL 60606 (US).

(81) Designated States (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,
KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME,
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,
OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,
SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ,
UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,
TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK,
EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,
MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
KM, ML, MR, NE, SN, TD, TG).

(54) Title: STORAGE MEMORY UNIT WITH A SHARED NONVOLATILE MEMORY INTERFACE FOR A RADIO

(57) Abstract: A storage radio unit (SRU) for a device can include a radio, embedded universal integrated circuit card (eUICC), a processor, an antenna, and nonvolatile memory. The SRU can support Micro SD card form factor standards and record a file system for a device. The device can be associated with a service provider and the SRU can be associated with a network provider. The radio can support Narrowband Internet of Things (NB-IoT) standards. The SRU can operate a file system interface (FSI) for the radio, where the device records application data in a file of the FSI. The SRU can attach to a wireless NB-IoT network using credentials recorded in the eUICC. The SRU can read the file of the FSI, and compress, encrypt, and transmit the application data to a network provider via the radio. The network provider can transmit the application data via TLS to the service provider.



WO 2020/061023 A1

STORAGE MEMORY UNIT WITH A SHARED NONVOLATILE MEMORY INTERFACE FOR A RADIO

BACKGROUND

Technical Field

The present methods and systems relate to enabling communications between a device and a wireless network, and more particularly, to operating a storage radio unit (SRU) in the device, where the SRU operates (i) nonvolatile memory, (ii) a shared nonvolatile memory interface with the device, and (iii) a radio unit for transmitting application data from the device to the wireless network using the shared nonvolatile memory interface.

Description of Related Art

The ability to connect transducers such as sensors and actuators with a network is a growing field with many economical applications. As the cost for both electronic hardware and bandwidth continues to decrease, the use of networked transducers is expected to continue increasing over the coming decades. Connecting transducers to a network can be referred to as “machine-to-machine (M2M) communications” or “the Internet of Things (IoT).” Among many potential benefits, IoT technologies allow automated monitoring and/or control of assets, equipment, personnel, or a physical location where manual monitoring may not be economical. Many applications for the “Internet of Things” significantly reduce costs by using automated monitoring instead of manual techniques. Prominent examples of IoT applications today include monitoring and control for building heating/air-conditioning, automobiles, alarm systems, and tracking devices. Fast growing markets for IoT applications today include health applications such as the remote monitoring of a person’s fitness activity, heartbeat, or glucose levels, monitoring of industrial equipment deployed remotely in the field, and also security systems.

IoT communications can provide remote control over actuators that may be connected to a device, such as turning on or off a power switch, locking or unlocking a door, adjusting a speed of a motor, or similar remote control. A decision to change or adjust an actuator associated with a connected device can utilize one or a series of sensor measurements. As one example, an IoT device for a truck can periodically report engine status to a remote server, and if the engine is operating outside specifications such as being too hot, including potentially an “alarm” condition, then temperature and an alarm code can be reported to a central server for the

IoT device. The server can subsequently instruct the driver and/or a specified mechanic to investigate the engine for potential mechanical malfunctions or other causes. The previous example is just one of many possible applications for IoT technology.

Many IoT applications can leverage wireless networking technologies, in addition to
5 wired technologies such as Ethernet. Wireless technologies such as wireless local area networks and wireless wide area networks have proliferated around the world over the past 20 years, and usage of these wireless networks is also expected to continue to grow. Wireless local area network (LAN) technologies include WiFi and wireless wide area network (WAN) technologies include 3rd Generation Partnership Project's (3GPP) 3rd Generation (3G) Universal Mobile
10 Telecommunications System (UMTS) and 4th Generation (4G) Long-term Evolution (LTE), LTE Advanced, Narrow-band Internet of Things (NB-IoT), as well as 5G networks. The many options to connect a transducer device to a network create opportunities for new products and services, but also create several classes of problems that need to be solved. Many of these problems or needs in the art pertain to conserving power for a device with transducers and
15 operating a radio, such that a battery in the device can have a long lifetime. A need exists in the art to enable a device to conserve power by minimizing the time that a processor and a radio in a device are in active states and maximizing the time the processor and radio in a device are in sleep states.

Many devices can utilize or require nonvolatile memory in order record data within the
20 device when the device is in a sleep, deep sleep, or even powered off state. The data required can include an operating system for the device. Additional data for a nonvolatile memory could be a logged record of transducer data, in order for the device to perform analysis or send summary information to a server. In addition, the device could continue to store transducer data for transmission at a subsequent time if a wireless network is not available. Consequently, a
25 device may need a nonvolatile memory storage unit such as a standard SD card or micro SD card. Many devices also connect with wireless WAN networks instead of WiFi networks, such as tracking devices that move, equipment that might frequently change location such as medical equipment in an ambulance, and devices monitoring the temperature or environment for food supplies, and other examples exist as well. Thus, many devices also need a wireless WAN radio
30 to connect with a wireless network, such as a network operated by a mobile network operator.

Conventional technology for IoT/M2M devices utilizes a separate storage unit (e.g. microSD card) and a separate radio module. A need exists in the art for a combined "storage radio unit", such that the storage radio unit includes both nonvolatile memory and a radio to connect with a wireless WAN. SD cards with WiFi radios and near field communication (NFC)

radios have been introduced into the market, and conventional technology for SD cards with radios supports local area networks. A need exists in the art for a microSD card to support data transfer through wireless WAN networks such as traditional mobile networks. A need exists in the art for a microSD card to support data transfer through NB-IoT wireless networking technology.

The wide variety of operating systems for devices with transducers creates significant challenges for a manufacturer of radio WAN modules for the devices. A radio module for a wide variety of devices may need to include many different versions of drivers in order to support the different operating systems. Developing and supporting the different drivers can create significant additional overhead or resource requirements for a manufacturer of radio modules. Therefore, a need exists in the art for a device and a storage radio unit to communicate with each other via established and widely deployed standards, such that new software drivers do not need to be created, distributed, and supported in order to enable communication between devices and a storage radio unit. A need exists in the art for a storage radio unit with a radio to communicate with a device using SD card drivers, since the vast majority of operating systems for devices will automatically support SD cards and related drivers.

Connecting a device to a wireless WAN such as supported by a mobile network operator traditionally requires the use of a Subscriber Identity Module (SIM) card. The use of a SIM card with an IoT/M2M device may not be convenient for many reasons, including lack of space, a desire for the device to be hermetically sealed, end users may prefer devices received to be “ready to connect” and not require additional handling of inserting a SIM card. Consequently, a need exist in the art for a storage radio unit that provides both nonvolatile memory and a WAN radio to also support an embedded universal integrated circuit card (eUICC).

The use of an eUICC for a storage radio unit can also create other challenges for users of devices. An eUICC according to GSMA M2M eUICC specifications requires a subscription manager secure routing (SM-SR) function in order to receive profiles, where a profile for the eUICC provides the equivalent of SIM card functionality. A device user or entity deploying the devices may not operate or support an SM-SR, which requires technical expertise and specialized server equipment, in addition to contractual agreements with mobile network operators and authorizations from the GSMA. Consequently, a need exists in the art for a storage radio unit with an eUICC to be provided with integrated support from an SM-SR, such that a device user could simply request an operational profile. A need exists in the art for the storage radio unit to include a bootstrap profile, such that the device automatically has connectivity wherever deployed. A need exists in the art for the SM-SR function included with

the storage radio unit to support changing the bootstrap profile to an operational profile based on requirements of the user for the device with the storage radio unit.

Narrowband wireless WAN technologies such as NB-IoT may not be able to efficiently use traditional IETF security protocols such as Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). A device operating at low power to conserve battery may have a few tens of kilobits per second of bandwidth and hundreds of milliseconds of round-trip delay with a server communicating with the device. A full DTLS handshake may require ~3KB and several seconds using conventional technology with a device connected to an NB-IoT network. A device may have only a hundred bytes or less to transmit in a report. Thus, conducting a DTLS session for each device report may waste considerable energy and bandwidth resources (e.g., > 90% of bandwidth and energy used for overhead). TLS would take longer and more bandwidth, given the additional overhead of TLS compared to DTLS.

A device may transmit reports an exemplary few times a day or less frequently, which can be outside the allowed session resume times for TLS and DTLS, and thus new TLS and DTLS sessions have to be established for each report. Consequently, a need exists in the art for the device or a storage radio unit to secure communication with a server in a manner that does not require standard TLS and DTLS. A need exists in the art for a server to convert the secured communications to standard security protocols such as TLS in order to communicate with an application server for the device, where the application server can support standard security protocols such as TLS.

And other needs exist in the art as well, as the list recited above is not meant to be exhaustive but rather illustrative.

SUMMARY

Methods and systems are provided for (i) operating a radio unit (RU) within a storage radio unit (SRU) and (ii) using a shared nonvolatile memory interface for communications between (a) a device operating the SRU and (b) the RU. The storage radio unit can comprise the form factor of a micro SD card or an integrated circuit soldered to a circuit board. The storage radio unit can also be referred to as a storage memory unit. A system supporting the communication for the device can include a wireless network, a SRU service provider, and a device service provider. The SRU service provider (SRU-SP) can operate a first server “server A”, and the device service provider can operate a second server “server B”. The wireless network, server A, and server B can be connected using the public Internet. The SRU-SP and the SRU can record a set of compatible cryptographic algorithms and cryptographic parameters, and the SRU-SP can also record at least one elliptic curve cryptography (ECC) private key. An objective of various embodiments of the disclosure is to address the needs in the art described above.

For an exemplary embodiment, a device can include electrical components of a processor, a transducer, random access memory (RAM), a battery, an antenna, a storage radio unit, and device service provider certificate, where the certificate includes a device service provider public key. The device can comprise a computing device to support M2M and IoT applications, in order to collect data regarding the state of a monitored unit and send the data to the second server, server B. The device can operate within range of a wireless wide area network, such as a network for a mobile network operator. The mobile network operator could support wireless narrowband technologies such as standards for “narrowband Internet of Things” (NB-IoT), which could be a subset of the “Long Term Evolution” LTE 4G standards. NB-IoT could also be supported by 5G networks as well.

Upon boot, the device can use an operating system with a driver for the storage radio unit, and libraries for (i) collecting data from a transducer, (ii) reporting data to a server in the form of application data, and (iii) writing the application data to the storage radio unit. The device can power the storage radio unit. The device can collect transducer data and write application data for server B to the storage radio unit. The processor for the device can enter and remain in a sleep state (a) after the device writes the application data to the storage radio unit until (b) SRU signals device 101 that the SRU has a server B response in a shared nonvolatile memory for the device to read. The processor for the device using this configuration for (i) entering and exiting the sleep state and (ii) using shared nonvolatile memory to read the server B response can conserve power. In other words, the processor can sleep for substantially

the entire time between (a) writing application data in a shared nonvolatile memory and (b) server B response data is available for reading by the processor in the device.

A storage radio unit can be manufactured with nonvolatile memory, a radio unit, an external bus, an internal bus, a memory processor, a memory core interface, RAM, an eUICC, a RF clock, EEPROM, a power control unit, and electrical connection to the antenna in the device. The nonvolatile memory can include shared memory for the device and the SRU, general memory for the device, and protected memory for the SRU and radio unit. Upon boot or power up of the SRU, the memory processor can load boot firmware from the EEPROM and then SRU firmware recorded in the protected memory for the SRU. After the boot process, the memory processor can subsequently communicate with (i) the device using the external bus and (ii) internal components in the SRU using the internal bus. The protected memory for the SRU can record an elliptic curve cryptography (ECC) SRU private key and an SRU-SP public key corresponding to the SRU-SP private key.

The radio unit in the SRU can include a processor, a RF front-end connected to the device antenna, a RF transceiver unit, and a baseband unit. For some embodiments, the RF-front-end can be external to the SRU and connected to the SRU via a data bus. The timing of the radio unit can be driven by the RF clock. The radio unit can be connected to the memory processor and the nonvolatile memory in the SRU using an internal bus. The radio unit can conduct a boot process and load RU firmware from the protected memory of the SRU. The power control unit in the SRU can control the RU and instruct the RU to enter a sleep state after boot.

The SRU firmware loaded by the memory processor can include libraries for the SRU to access and control nonvolatile memory in the SRU via the memory core interface and the internal bus. The SRU firmware can also include a configuration process. The configuration process can read and detect the type of filesystem written to the general memory for the device. The configuration process can use filesystem drivers in the SRU firmware to configure the shared nonvolatile memory with the detected filesystem. The configuration process can write a device write and SRU read directory in the shared nonvolatile memory. The configuration process can write a SRU write and device read directory in the shared nonvolatile memory. The device write directory and the SRU write directory can comprise a shared filesystem interface between the device and the SRU. The configuration process can then load an initial set of files in the shared nonvolatile memory. The operating system for the device can mount the filesystem for the shared nonvolatile memory and the device general memory. The device can

access the shared nonvolatile memory and the device general memory using a standard driver for the operating system of the device, such as an SD card driver.

The device can write the application data to the device write directory in the shared filesystem interface operating in the nonvolatile memory of the SRU. The memory processor
5 can process a trigger library from the SRU firmware to detect that application data has been written to the shared filesystem interface. The trigger library can instruct the power control unit to change the state of the radio unit from a sleep state to an active state. Upon entering the active state, the radio unit can attach to the wireless network using credentials recorded in a profile for the eUICC. The radio unit (RU) can then read the application data from the device
10 write directory.

The RU can derive an ECC ephemeral PKI key pair comprising an ephemeral private key and an ephemeral public key. The RU can conduct an ECDH key exchange using as input the ephemeral private key, the SRU private key, and the SRU-SP public key. The output of the ECDH key exchange can comprise a secret shared key that also can be mutually derived by
15 server A of the SRU-SP. The secret shared key can be input into a key derivation function in order to derive a symmetric ciphering key. The RU can compress the application data using a compression library included in SRU firmware. The RU can encrypt the compressed application data using the derived symmetric ciphering key and a symmetric ciphering algorithm. The RU can transmit a message to server A through the narrowband or wideband wireless network
20 comprising (i) the encrypted, compressed application data along with (ii) the derived ephemeral public key and (iii) an identity of the SRU.

Server A of the SRU-SP can record an SRU public key corresponding to the SRU private key recorded by the SRU. Server A can receive the message from the RU via the wireless network and the Internet. Server A can read the SRU identity in the message and lookup the
25 SRU public key as well as cryptographic parameters associated with the SRU and SRU public key. For some embodiments, the SRU identity can comprise a secure hash value over a static public key for the SRU. The SRU can read the ephemeral public key received in the message. The SRU-SP can conduct an ECDH key exchange using the received ephemeral public key, the SRU public key, and the SRU-SP private key in order to derive the secret shared key. The SRU-
30 SP can input the secret shared key into a key derivation function in order to derive the symmetric ciphering key. The SRU-SP can decrypt the encrypted, compressed application data using the derived symmetric ciphering key and a symmetric ciphering algorithm. The SRU-SP can decompress the application data in order to read the plaintext application data. The SRU-SP

can establish a TLS session with server B. The SRU-SP can forward the application data in the TLS session to server B.

Server B can receive the application data for the device. The application data can include a device identity. Server B can process the data and generate a server B response along with a digital signature for the server B response data using a SRU-SP private key. Server B can send Server A for the SRU-SP the server B response and the digital signature for the server B response data through the TLS session. Server A can map or track which sessions with server B are associated with which SRU via the SRU identity received in the message from the RU. The SRU-SP can receive the server B response and compress the data. The SRU-SP can encrypt the compressed data using the derived symmetric ciphering key. The SRU-SP can forward (a) the encrypted, compressed server B response data and (b) the digital signature for the server B response data to the RU via the Internet and the narrowband or wideband wireless network.

The RU in the SRU can receive (a) the encrypted, compressed server B response data and (b) the digital signature. The RU can decrypt the encrypted, compressed server B response data using the derived symmetric ciphering key and the symmetric ciphering algorithm. The RU can decompress the compressed server B response data. The RU can write (a) the server B response data and (b) the digital signature to the SRU write and device read directory in the shared nonvolatile memory. The power control unit can (i) return the RU to a sleep state and (ii) signal the device for the device to wake the processor in the device or change the processor in the device from the sleep state to the active state. The device can read the server B response data and the digital signature for the server B response data from the shared nonvolatile memory. The server B response data could include an actuator setting or other command for the device, or simply an acknowledgement that the application data was received. The device can verify the digital signature using the certificate recorded for the device service provider (which operates server B). After reading and processing the server B response data, the device can optionally delete the data from the shared nonvolatile memory.

For another exemplary embodiment, a device can include the radio unit and nonvolatile memory, without requiring the use of a storage radio unit. The device or a configuration unit can conduct a configuration process on the nonvolatile memory to configure a shared nonvolatile memory with a device write memory and a RU write memory. The device can write application data to the device write memory. The device can operate a power control unit and change the state of the RU from a sleep state to an active state after the device writes the application data. A processor in the device can then enter a sleep state until the radio unit writes server B response data to the RU write memory. The RU can wake from a sleep state and

connect with a wireless network using an eUICC. The RU can read the application data from device write memory and conduct the steps for a RU listed above for the RU to transmit and send the application data.

Server A and server B can conduct the steps listed above for receiving and processing the application data. Server A and server B can conduct the steps for processing and communicating the server B response data and a digital signature from server B for the server B response data. The RU in the device can (i) receive the server B response data and digital signature via the wireless network and (ii) write the server B response data and the digital signature to the RU write memory. The RU in the device can signal the processor in the device to exit from a sleep state in order for the processor in the device to read the server B response data and the digital signature. The device can then verify the digital signature over the server B response data using a recorded certificate associated with server B. The device can then process and operate with the server B response data.

For another exemplary embodiment, a device can include a system on a chip, where the system on the chip includes a processor, nonvolatile memory, and a radio unit. The components in the system on the chip can be connected via an internal system bus. For embodiments with the “system on a chip” configuration, some components for an RU used in a storage radio unit configuration may be operated outside the system on a chip, such as an RF front end operated outside the system on a chip. An RF clock within the system on the chip can drive the internal timing of both the processor and the internal radio units. The nonvolatile memory can comprise a persistent memory or a nonvolatile RAM (NVRAM) memory. The system on a chip can operate with the steps for a device according to the steps for a device and a RU listed above in the previous two paragraphs, where (i) the processor writes application data for the RU to the nonvolatile memory in the system on a chip and (ii) the RU writes server B response data and a digital signature for the processor to the nonvolatile memory in the system on a chip.

The processor can signal the RU to wake after writing the application data and also enter a sleep state. The RU can signal the processor to wake after writing the server B response data and the digital signature, and also return to a sleep state. The processor can read the server B response data and the digital signature, verify the digital signature using a certificate associated with server B, and process and operate with the server B response data. In some exemplary embodiments, the server B digital signature can be omitted, and the processor can conduct encryption of the application data before writing the application data for the RU to the shared nonvolatile memory.

These as well as other aspects and advantages will become apparent to those of ordinary skill in the art by reading the following detailed description, with reference where appropriate to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

5 Various exemplary embodiments are described herein with reference to the following drawings, wherein like numerals denote like entities.

Figure 1a is a graphical illustration of an exemplary system, where a storage radio unit operates with a device, and the storage radio unit communicates with a service provider using a radio and a wireless network, in accordance with exemplary embodiments;

10 Figure 1b is a graphical illustration of hardware, firmware, and software components for a device, including a storage radio unit, in accordance with exemplary embodiments;

Figure 1c is a graphical illustration of hardware, firmware, and software components for a device and a storage radio unit, in accordance with exemplary embodiments;

15 Figure 2a is a graphical illustration of components within a storage radio unit, in accordance with exemplary embodiments;

Figure 2b is a graphical illustration of an exemplary system, where a device and a radio unit communicate through a shared nonvolatile memory in a storage radio unit, in accordance with exemplary embodiments;

20 Figure 2c is a graphical illustration of a storage radio unit utilizing two clock inputs, in accordance with exemplary embodiments;

Figure 2d is a graphical illustration for using an interface between a device and a storage radio unit, in accordance with exemplary embodiments;

Figure 2e is a graphical illustration of storage radio unit firmware 109u, in accordance with exemplary embodiments;

25 Figure 2f is a flow chart illustrating exemplary steps for configuring a storage radio unit to communicate with a device using a shared nonvolatile memory interface, in accordance with exemplary embodiments;

Figure 3a is a graphical illustration of a shared filesystem interface for a radio storage unit and a device, using a device write directory, in accordance with exemplary embodiments;

30 Figure 3b is a graphical illustration of a shared filesystem interface for a radio storage unit and a device, using a radio unit write directory, in accordance with exemplary embodiments;

Figure 3c is a flow chart illustrating exemplary steps for a device and a storage radio unit to utilize a shared filesystem interface, in accordance with exemplary embodiments;

Figure 4a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a storage radio unit, a radio unit, a server A, and a server B, in accordance with exemplary embodiments;

5 Figure 4b is a flow chart illustrating exemplary steps for conducting a key exchange using PKI keys in order to derive a shared secret key and a symmetric ciphering key, in accordance with exemplary embodiments;

Figure 5a is a graphical illustration of hardware, firmware, and software components for a device and a storage radio unit, including a shared memory interface operating in RAM for the storage radio unit, in accordance with exemplary embodiments;

10 Figure 5b is a flow chart illustrating exemplary steps for configuring a storage radio unit to communicate with a device using a shared nonvolatile memory interface in RAM, in accordance with exemplary embodiments;

Figure 6a is a flow chart illustrating exemplary steps for storage radio unit to communicate with a device using a shared nonvolatile memory interface, in accordance with
15 exemplary embodiments;

Figure 6b is a flow chart illustrating exemplary steps for a storage radio unit to operate a radio unit and a shared nonvolatile memory interface, in accordance with exemplary embodiments;

Figure 7 is a graphical illustration of hardware, firmware, and software components for a
20 device, in accordance with exemplary embodiments; and,

Figure 8 is a graphical illustration of hardware, firmware, and software components for a device, in accordance with exemplary embodiments.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS OF THE INVENTION**Figure 1a**

Figure 1a is a graphical illustration of an exemplary system, where a storage radio unit operates with a device, and the storage radio unit communicates with a service provider using a radio and a wireless network, in accordance with exemplary embodiments. The system 100 includes a device 101, a wireless network 102, a storage radio unit service provider (SRU-SP) 106, and a device service provider 110. Device 101 can include a storage radio unit (SRU) 109 and an antenna 101t for connecting to the wireless network 102. Device 101 can be a wireless computing device to support applications for the “Internet of Things” or “Machine-to-Machine” communications including tracking devices, security systems, cameras, or health monitoring devices. Device 101 can also be a computing device, a mobile phone, a module, a drone, a laptop computer, a desktop computer, a server, a tablet, a “point of sale” cash register or payment terminal, and many other possibilities exist as well. Device 101 can be associated with or operate in conjunction with a monitored unit 105. Exemplary electrical and physical components operating within device 101 are also depicted and described in connection with Figure 1b and Figure 1c below. Although a single device 101 is depicted in Figure 1a, a wireless network 102 could include a plurality of devices 101.

Device 101 can be associated with a monitored unit 105. Examples of a monitored unit 105 can include an ATM or vending machine, a truck or automobile, a refrigerated or standard (“dry”) shipping container, or industrial equipment such as, but not limited to, an oil field pump, a transformer connected to an electrical grid or an escalator or elevator in a building. Additional examples of a monitored unit 105 include can also include a pallet for shipping or receiving goods, a refrigerator with food, a health monitoring device attached to a person such as, but not limited to, a heart monitor, and a gate or door for opening and closing. Device 101 can utilize a sensor (depicted in Figure 1b) to measure and collect data regarding a parameter of monitored unit 105 such as, but not limited to, temperature, humidity, physical location potentially including geographical coordinates from a Global Positioning System (GPS) or other satellite system receiver, surrounding light levels, surrounding RF signals, vibration and/or shock, voltage, current, audio or sound, and/or similar measurements. Other possibilities exist as well for device 101 to associate with a monitored unit 105 without departing from the scope of the present disclosure. In some exemplary embodiments, a monitored unit 105 could be omitted,

and data from a device 101 directly could be communicated through wireless network 102 to a server such as server A 108 and server B 110a.

Monitored unit 105 could also be controlled by device 101 via a transducer that is an actuator (depicted in Figure 1b), where device 101 can change the actuator in order to change a state or parameter for monitored unit 105. For example, if monitored unit 105 is a door, then a transducer for device 101 could include a relay to activate a lock for the door. If monitored unit 105 is a lighting system in a building, then a transducer could comprise a switch to turn the lighting level up or down. Other examples exist for monitored unit 105 as well, and the above are intended to be illustrative instead of limiting. Device 101 could forward sensor data to device service provider 110 as application data 112a. Device service provider 110 can both (i) receive application data 112a from sensors in device 101 for monitored unit 105 and (ii) transmit server B response data 112b for actuators in device 101 in order to control monitored unit 105. Thus, in operation and by using the systems and methods described in subsequent figures below, device service provider 110 can remotely monitor and control monitored unit 105 using device 101, where device 101 obtains connectivity to wireless network 102 via SRU 109, or a radio unit operating in device 101.

Wireless network 102 can comprise a wide area wireless network operated by a mobile network operator and support wireless wide area network (WAN) technologies including 3rd Generation Partnership Project's (3GPP) 3rd Generation (3G) Universal Mobile Telecommunications System (UMTS) and 4th Generation (4G) Long-term Evolution (LTE), LTE Advanced, Narrow-band Internet of Things (NB-IoT), and 5G standards as well, including subsequent standards. Wireless network 102 could also support other wireless standards such as low power wide area networking such as Sigfox and LoRa. Wireless network 102 can operate a base station 103, which could comprise an "eNodeB" (for 4G networks), a "gNodeB" (for 5G networks), a picocell, a macrocell, or other wireless WAN radio access network technologies. For some exemplary embodiments, base station 103 could comprise an access point for an 802.11 network, such as, but not limited to, and 802.11 ac, 802.11 ax, or 802.11 ah network.

Wireless network 102 can require device 101 to record and operate with (i) a set of network access credentials such as an international mobile subscriber identity (IMSI) and key K for 4G networks, and/or (ii) a subscription permanent identity (SUPI) and master key for 5G networks, and other possibilities exist as well. Network access credentials for wireless network 102 can be recorded in an embedded universal integrated circuit card (eUICC) 113 in SRU 109 in device 102. Although a single wireless network 102 is depicted in Figure 1a, a system 100 could include a plurality of wireless networks 102. In addition, wireless network 102 could

support WiFi wireless networking, including, but not limited to, 802.11 based standards, such as “ac”, “ah”, and future or related versions of WiFi standards as well including 802.11 ax.

Device 101 can communicate with SRU-SP 106 via the wireless network 102. SRU-SP 106 can comprise an entity, business, or collection of computing devices associated with SRU 109 and providing data communications services to device 101 via SRU 109. SRU-SP 106 can include a Subscription Manager – Secure Routing (SM-SR) 107 function according to GSM Association standard SGP.02 version 3.2 titled “Remote Provisioning Architecture for Embedded UICC Technical Specification”, which is herein incorporated by reference in its entirety. SM-SR 107 can record information pertaining to eUICC 113 in SRU 109, including an eUICC information set (EIS). SM-SR 107 could also communicate with a plurality of different Subscription Manager – Data Preparation (SM-DP) functions (not shown), where an SM-DP could be associated with wireless network 102. The SM-SR 107 can establish a secure channel with the eUICC 113 and transfer profiles to the eUICC through the secure channel, according to the specification in GSMA SGP.02. The SM-SR 107 can receive profiles for eUICC 113 from a SM-DP, based on requests or contractual agreements between the SM-SR 107 and the SM-DP.

SRU-SP 106 can also include at least one server A 108 for communicating application data 112a and server B response 112b data with SRU 109 via the wireless network 102. Server A 108 can comprise a server in order to communicate with a plurality of SRU 109 in different devices 101. Different devices 101 could operate with different wireless networks 102. Server A 108 and server B 110a for both SRU-SP 106 and device service provider 110 can be either different physical computers such as rack-mounted servers, or different logical or virtual servers or instances operating in a “cloud” configuration. Internal electrical and physical components for server A 108 and server B 110a can be equivalent to those depicted and described for a device 101 in Figure 1b and Figure 1c below, but with high capacities associated with servers. Server A 108 and server B 110a for both SRU-SP 106 and device service provider 110 can each be co-located within the same data center or geographically dispersed.

Server A 108 and server B 110a for both SRU-SP 106 and device service provider 110 can operate with an operating system for a server such as Linux or Windows Server, and other possibilities exist as well. Although a single server A 108 and server B 110a is depicted in Figure 1a, server A 108 and server B 110a could each separately comprise a plurality of different servers or computing devices with different functions to operate collectively as a single server, such as a firewall, a database, a router, and an application such as a web server or an email server. In addition, although server A 108 and server B 110a are depicted as operating

with separate entities, the function of server A 108 and server B 110a could be combined into a single server operating with one of the two entities SRU-SP 106 or device service provider 110.

Server A 108 can include a SRU public key 109w for each SRU 109, a device service provider certificate 110c for device service provider 110, and an SRU-SP private key 106a. SRU public key 109w can be selected based on an identity for SRU 109, and SRU public key 109w can also be recorded in a certificate. Certificates in the present disclosure can be formatted according to the X.509v3 and related standards, and record information about the public key, unit holding the corresponding private key, and cryptographic parameters associated with the public and private keys. In exemplary embodiments, public and private keys can be used with cryptographic algorithms supporting key exchange and digital signatures. Public and private keys can support either elliptic curve cryptography (ECC) or RSA-based cryptographic algorithms, and other cryptographic algorithms with public and private keys can be supported as well, including cryptographic algorithms and PKI keys supporting post-quantum cryptography, without departing from the scope of the present disclosure.

Device service provider 110 can comprise an entity, business, or collection of servers providing service to device 101. Device service provider 110 can include device service provider private key 110b, server B 110a, and configuration unit 110d. Device service provider private key 110b can comprise a private key corresponding to device service provider public key 110c (or certificate) recorded by SRU-SP 106 in server A 108. Server B 110a can comprise a server or computing device to receive data from device 101 over an IP network 111 and can also include data regarding monitored unit 105. Although not depicted in Figure 1a, server B 110a can include a database recording application data 112a for a plurality of different devices 101.

Server B 110a can also function as a monitoring and controlling service for device 101, such as logging data and also sending commands or instructions to device 101 in server B data 112b. Server B 110a could support a wide variety of applications, such as tracking location of device 101, monitoring physical security status reported by device 101, monitoring health data for a user associated with device 101, monitoring the status or state of monitored unit 105, and other possibilities exist as well without departing from the scope of the present disclosure. In exemplary embodiments, server B 110a can be referred to as an application server. Server B 110a could support web interfaces to users for users or owners of device 101 to monitor the status of device 101. Although Figure 1a depicted server B data 112b for device 101 as being transmitted through SRU-SP 106, in some embodiments application data 112a and server B data 112b could be communicated between device 101 and device provider 110 without passing through SRU-SP 106.

In some embodiments, device service provider 110 can operate both server A 108 and server B 110a. For these embodiments, SRU-SP 106 could continue to operate SM-SR 107 but not operate server A 108. For these embodiments, device service provider 110 obtains the benefits for the use of a server A 108, where communication and encryption of data between server A 108 and RU 109b can comprise a first protocol optimized for an NB-IoT 104 network or other narrowband wireless network technology. Communication between server A 108 and server B 110a can utilize standard IETF security protocols such as TLS between the two servers. In other words, TLS or DTLS may not be optimal or preferred for the bandwidth and power/energy requirements of device 101 with RU 109b operating on a battery and with narrowband wireless technologies. Consequently, a first server comprising server A 108 can manage the communication with RU 109b over the narrowband wireless technology using a first protocol for communication and security, and server A 108 can manage the communication and security with server B 110a using a second protocol for communication and security, such as TLS. Additional details regarding communication between RU 109b, server A 108, and server B 110b are provided below in Figure 4b and related figures.

Server B 110a can include an Internet Protocol (IP) address 110e, which could comprise an address according to IPv4 or IPv6 standards. Other servers and computing devices depicted in Figure 1a could include an IP address as well and connect to an IP network 111. Data between the nodes and entities depicted in Figure 1a can be communicated via the IP network 111 using IP addresses such as IP address 110e. IP network 111 can comprise either the globally routable public Internet or also a private network which could be an intranet. IP network 111 could comprise a combination of intranets linked to the globally routable public Internet.

Configuration unit 110d for device service provider 110 can comprise a device for (i) configuring device nonvolatile memory 109a in SRU 109 and (ii) recording firmware, software, and data for device 101 in the configured memory 109a. By inserting SRU 109 in configuration unit 110d before SRU 109 is inserted into device 101, memory 109a can record data for device 101 before device 101 begins operation. Although not depicted in Figure 1a (but depicted and described in connection with Figure 1c and Figure 2a below), (x) memory 109g for device 101 in SRU 109 can reside within nonvolatile memory 109a in SRU 109, and (y) a configuration unit 101d could configure memory 109g in memory 101a. The use of a configuration unit 110d can be omitted in some embodiments, and a device 101, user or SRU 109 manufacturer can configure memory 109g for those embodiments. Although a configuration unit 110d is depicted

in Figure 1a as associated with a device service provider 110, for some exemplary embodiments a SRU-SP 106 could operate a configuration unit 110d.

Storage radio unit 109 can comprise a computing device that can be inserted into device 101. In exemplary embodiments, storage radio unit (SRU) 109 can comprise a Secure Digital (SD) card supporting physical standards maintained by the SD Card Association (SDA). SRU 109 can have a physical form compatible with (i) standard SD cards and dimensions of 32 mm x 24 mm x 2.1 mm, or (ii) micro SD card and dimensions of 15 mm x 11 mm x 1.0 mm. Other possibilities exist as well for the physical form of SRU 109, including the “miniSD” size. SRU 109 could also comprise an integrated circuit that could be soldered onto a circuit board for device 101. SRU 109 could also comprise a wafer level chip scale package (WLCSP), which could be included in a separate integrated circuit, such as a “system on a chip” for device 101 (as depicted in Figure 8 below). Note that in some exemplary embodiments, the components depicted for storage radio unit 109 could be operated by other electrical components within device 101, such as depicted in Figure 7 and Figure 8 below, and in this case an SRU 109 could be omitted while obtaining the benefits and functionality of the present disclosure, including the use of a shared nonvolatile memory interface for radio unit 109b and device 101 as described below in Figure 2b, Figure 2f, Figure 3a, Figure 3b, etc.

Storage radio unit 109 can include nonvolatile memory 109a, radio unit 109b, SRU private key 109p, SRU-SP public key 106b, and eUICC 113. In exemplary embodiments, each of nonvolatile memory 109a, radio unit 109b, SRU private key 109p, and eUICC 113 comprise either (i) wafer level chip scale packages (WLCPS) or (ii) separate components etched and processed within a single silicon chip or package, in order to support a micro SD card form factor. SRU private key 109p can comprise the private key corresponding to SRU public key 109w recorded with SRU-SP 106 and described in this Figure 1a above. SRU-SP public key 106b can be the public key corresponding to the SRU-SP private key 106a recorded and operated by server A 108 with SRU-SP 106.

Non-volatile memory 109a can comprise a physical memory of NAND or NOR flash memory, such that data recorded in non-volatile memory 109a continues to be recorded when electrical power is removed from storage radio unit 109. The data within non-volatile memory 109a can subsequently be read and/or re-written when power is restored to storage radio unit 109. Exemplary values for the amount of memory available non-volatile memory 109a for SRU 109 operating as an SD card physical form factor can range from 512 megabytes to 128 gigabytes, and other possibilities exist as well. For some embodiments, nonvolatile memory

109a can comprise a persistent memory or nonvolatile RAM memory (NVRAM), such as that depicted in Figure 8 below.

Non-volatile memory 109a can include occasional bit errors due to the nature of the physical memory, such as small memory cell sizes, but error correcting codes operating in a memory processor in SRU 109 (depicted as processor 109d in Figure 1c below) can normally correct the errors, such as limiting errors in a file read by device 101 from non-volatile memory 109a to less than one part per billion during the normal, operating lifetime of storage radio unit 109. As contemplated herein, error correcting codes can comprise either (i) convolution codes operating on a bit by bit basis on memory or data from memory, such as a Viterbi decoder, or (ii) block codes, such as a Hamming codes or Reed-Solomon codes. Other possibilities for the physical structure of non-volatile memory 109a and error correcting codes exist as well without departing from the scope of the present disclosure, and generally (i) non-volatile memory 109a includes addresses and blocks, such that binary data can be recorded and subsequently read in a nonvolatile state, and (ii) error correcting codes attempt to identify and correct the presence of bit errors in either physical memory and/or data read from the physical memory.

Radio unit 109b operating in SRU 109 can comprise a radio with standard radio components such as RF filters, RF amplifiers, a clock, a controller or processor, and phased loop logic (PLL), and may be connected with an antenna such as antenna 101t for device 101. Additional details for radio unit 109b are depicted and described in connection with Figure 2a below. In exemplary embodiments, radio unit 109b supports the wireless WAN standards for wireless network 102, which could comprise a network supporting NB-IoT 104 radio frequency signaling and standards. NB-IoT was initially standardized with 3GPP Release 13, and large mobile networks in the United States such as T-Mobile and AT&T are deploying NB-IoT networks nationwide throughout 2018 and 2019. Benefits of NB-IoT 104 communications for radio unit 109b include low power and high sensitivity, such as using sub-channels or carriers totaling ~180 kHz and bit rates on the order of ~100 kbps, or lower for “deep penetration” in buildings or remote locations. Other narrowband wireless technologies could be used with RU 109b and wireless network 102 as well.

In exemplary embodiments, radio unit 109b supports subsequent and related standards for NB-IoT 104, including updates through 3GPP Release 16, which is the latest in formal draft phase as of 2018. Radio unit 109b could also support other relatively low electrical power and low processing power standards or protocols, including Sigfox and Lora. In general, the radio protocols supported by radio unit 109b are characterized by (i) low transmit power such as a maximum transmit power of 30 dBm or less, and (ii) relatively narrow frequency channels such

as less than 2 MHz, and preferably approximately equal to less than 200 kHz. For NB-IoT applications, narrow sub-channels or carriers such as 15 or 3 kHz could be utilized. For some embodiments, wireless network 102 and RU 109b could support LTE CAT M1, which comprises using approximately 1 MHz of RF spectrum. RU 109b could operate in either half duplex mode or a full duplex mode. The RF bands for radio unit 109b could use either frequency division duplex (FDD) or time division duplex (TDD).

eUICC 113 operating with SRU 109 can operate according the GSMA standards in SGP.02 for an eUICC, including subsequent and related standards. eUICC 113 can be manufactured by an eUICC Manufacturer (EUM) certified by the GSMA. eUICC 113 can include an Issuer Security Domain – Root (ISD-R) and an eUICC Controlling Authority Security Domain (ECASD), along with an internal processor and memory and a bus. eUICC 113 could also comprise a secondary platform bundle operating within a “Smart Secure Platform” within SRU 109. Separate profiles can be recorded in Issuer Security Domains – Profile (ISD-P), which can be created based on commands from the SM-SR 107 and SM-DP associated with wireless network 102. A profile can contain the data equivalent to that recorded in a traditional SIM card comprising network access credentials (e.g. IMSI and key K) and wireless network 102 parameters (e.g. frequencies or channels to use, network codes, timers to implement, etc.)

In exemplary embodiments, eUICC 113 includes a bootstrap profile 113c and an operational profile 113d. Bootstrap profile 113c can be written to eUICC 113 by the SM-SR 107 function, or specified by SM-SR 107 and written by the EUM. For some embodiments, eUICC 113 could use GSMA standards SGP.22 for consumer devices (where device 101 can include a user interface 101j as depicted in Figure 1b below). Use of SGP.02 may be preferred if device 101 does not operate a user interface. In exemplary embodiments, eUICC 113 can support both SGP.02 and SGP.22.

Bootstrap profile 113c provides initial or fallback connectivity on a global basis, and could be associated with a globally operating wireless network 102 such as AT&T or Vodafone. Operational profile 113d could be downloaded or updated using the steps specified for a profile download procedure with an eUICC from GSMA standards. Operational profile 113d could be used by SRU 109 and device 101 based on updated contractual agreements with a wireless network 102 or for use in specific geographies. Although a single eUICC 113 is depicted for SRU 109 in Figure 1a, in some exemplary embodiments an SRU 109 could include multiple different eUICC 113 and select the eUICC 113 based on preferences in SRU 109 firmware (such as SRU firmware 109u in Figure 1c, 2a, 2e, etc. below). Benefits of SRU 109 containing

multiple eUICC 113 units is that SRU 109 can select a profile to use from each of the multiple different eUICC 113 based on the operational profile 113d in each of the different eUICC 113.

According to the current GSMA specifications for eUICCs in embedded devices (such as with SGP.02), an operational profile 113d cannot be changed without a command from or interaction with an SM-DP (except for fallback or bootstrapping cases). By using multiple different eUICC 113 in SRU 109, the SRU 109 can select a different eUICC 113 with different operational profiles 113d in each eUICC 113 in order to have access to multiple profiles without involving an SM-DP. The selection of an eUICC 113 by SRU 109 is described with step 406a in Figure 4 below. An eUICC 113 can also be referred herein and within related literature as an “integrated universal integrated circuit card” (iUICC), as depicted with Figure 8 below. In exemplary embodiments, the same SM-SR 107 could be used with each of the different eUICC 113 operating in device 101. Or, in some embodiments a single eUICC can be used with SRU 109 and device 101.

Using the components depicted for a system 100, a device 101 with an SRU 109 and a radio unit 109b can connect with a first wireless WAN network 102, such as using NB-IoT 104 standards. Wireless WAN network 102 can provide access to IP network 111 and SRU-SP 106 and/or device service provider 110. The SM-SR 107 function of SRU-SP 106 can support sending an operational profile 113d to the eUICC 113 operating in SRU 109. SRU 109 can use the operational profile 113d to connect with a second wireless network 102, based on the network access credentials and network parameters in the downloaded operational profile 113d. Device 101 can then send application data 112b to SRU-SP 106 via the second wireless network 102 and IP network 111. SRU-SP 106 can establish a secure session with device service provider 110 such as TLS session 114. SRU-SP 106 can then forward the application data 112a to device service provider 110 via a TLS session 114.

The device service provider 110 can receive and process the application data 112a from the TLS session 114 using server B 110a. The device service provider 110 can send server B response 112b data to SRU-SP 106 via the TLS session 114. The SRU-SP 106 can then forward server data 112b for device 101 via the second wireless network 102 and the SRU 109 operating within device 101. Server B response 112b data could be a response or confirmation or command for device 101 based on the application data 112a received by device service provider 110. SRU 109 can write the server B response 112b to an interface operating in nonvolatile memory 109a, which can be subsequently read by device 101. In this manner, device 101 can use an SRU 109 to communicate with device service provider 110.

Note that the use of SRU-SP 106 can support standard IETF protocols such as TLS version 3 (e.g. draft-ietf-tls-tls13-28) through IP network 111 with device service provider 110. However, narrowband wireless network technologies such as NB-IoT may not be suited for TLS due to the low bandwidth and potential higher latency (for “deep” building penetration or remote locations). For example, device 101 may prefer not to establish a TLS or DTLS session with server B 110a each time device 101 sends small reports or data such as an exemplary ~50 or ~100 bytes. Transmitting small sets of application data 112a to server B 110a (via wireless network 102 with NB technology and server A 108) may be better supported including a longer battery life for device 101 through a system 100. Also note that application data 112a can be compressed and encrypted by SRU 109 before transmission to SRU-SP 106 and server data 112b can be decrypted and decompressed by SRU 109 after receipt from SRU-SP. In this manner, a system 100 can address needs in the art described in the background above. Additional details regarding the operation of a system 100 and components will be provided in figures below.

Figure 1b

Figure 1b is a graphical illustration of hardware, firmware, and software components for a device, including a storage radio unit, in accordance with exemplary embodiments. Figure 1b is illustrated to include many electrical components that can be used by device 101. Device 101 may also operate in a wireless configuration in order to connect with a wireless network 102. Device 101 may consist of multiple components in order to collect sensor data or control an actuator associated with a monitored unit 105. The physical interface 101p can include associated hardware to provide connections to components such as user interface 101j, a power supply, external transducers, a display or light-emitting diodes, etc. Device drivers 101g can communicate with the physical interfaces 101p, providing hardware access to higher-level functions on device 101. Device drivers 101g may also be embedded into hardware or combined with the physical interfaces.

Device drivers 101g can include a SD card driver 101k, which can be utilized by a device 101 and operating system 101h in order to read and write data to storage radio unit 109, including communicating with a radio unit 109b within storage radio unit 109. SD card driver 101k can operate (i) an SD bus or (ii) a serial peripheral interface (SPI) for physical interface 101a with SRU 109. Other device drivers and data busses than supporting an SD card can be utilized by a device 101 as well in order to communicate with an SRU 109. Additional details

regarding the communication between device 101 and radio unit 109b within storage radio unit 109 are described in additional figures below, including Figure 2b and Figure 2d. Device 101 may preferably include an operating system 101h to manage device drivers 101g and hardware resources such as physical interface 101p and RAM 101e within device 101. The operating systems described herein can also manage other resources such as processor 101b and may support multiple software programs or software libraries operating on device 101.

The operating system 101h can include Internet protocol stacks such as a User Datagram Protocol (UDP) stack, Transmission Control Protocol (TCP) stack, a domain name system (DNS) stack, etc., and the operating system 101h may include timers and schedulers for managing the access of software to hardware resources, including storage radio unit 109. The operating system shown of 101h can be appropriate for a low-power device with limited memory and CPU resources (compared to a server 106 or 110a). Example operating systems 101h for a device 101 includes Linux, Android® from Google®, iOS from Apple®, Windows® Mobile, or Open AT® from Sierra Wireless®. Additional example operating systems 101h for device 101 include eCos, uC/OS, LiteOs, Contiki, OpenWRT, Raspbian, and other possibilities exist as well without departing from the scope of the present disclosure.

A device program 101i may be an application programmed in a language such as C, C++, Java, and/or Python, and could provide functionality to support M2M applications such as monitoring of sensors and activation or control of actuators. Device program 101i could also be a software routine, subroutine, linked library, or software device, according to one preferred embodiment. As contemplated herein, a device program 101i may be an application operating within a smartphone, such as an iPhone® or Android®-based smartphone, and in this case device 101 could comprise the smartphone. The application functioning as a device program 101i could be downloaded from an “app store” associated with the smartphone or operating system for the device 101. Note that the use of a smartphone for device 101 is not required for some exemplary embodiments.

Device program 101i can include data reporting libraries 101x, which can provide the functionality or CPU 101b instructions for (i) collecting sensor data, and (ii) sending messages such as application data 112a to SRU 109 via a structure of device write directory 191x'. In other words, a structure of device write directory 191x' is different than a device write directory 191x (from Figure 3a below), where device write directory 191x could reside in a shared nonvolatile memory 101h, while the structure of device write directory 191x' in a data reporting libraries 101x can comprise instructions, structure, and format such that data reporting libraries 101x can write to a device write directory 191x in memory 109h.

Similarly, data reporting libraries 101x can store a structure of RU write directory 192x', where a structure is stored in 101x and the actual RU write directory 192x can reside in shared nonvolatile memory 109h. Data reporting libraries 101x can provide CPU 101b instructions for (i) reading server B response 112b data from shared nonvolatile memory 109h. Device program 101i can also include the logic for connecting with a server A 110a (via SRU 109), sending a device identity 101a to the server A 110a, receiving a challenge or nonce from server A 110a, and sending a digital signature for the challenge or nonce to the server, where the digital signature can be calculated using a certificate 110c recorded in device 101.

Note that device 101 may also optionally include user interface 101j which may include one or more devices for receiving inputs and/or one or more devices for conveying outputs. User interfaces are known in the art and generally are simple for devices such as a few LED lights or LCD or OLED display, and thus user interfaces are not described in detail here. User interface 101j could comprise a touch screen if device 101 operates as a smartphone or mobile phone. As illustrated in Figure 1b, device 101 can optionally omit a user interface 101j, since no user input may be required for many M2M applications, although a user interface 101j could be included with device 101.

Device 101 may be a computing device that includes computer components for the purposes of collecting data from a sensor 101f or triggering an action by an actuator 101y. Device 101 may include a central processing unit (CPU) 101b, a random access memory (RAM) 101e, and a system bus 101d that couples various system components including the random access memory 101e to the processing unit 101b. The system bus 101d may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures including a data bus. Device 101 can operate a plurality of different busses 101d such as a memory bus, an I2C bus, a USB bus, a SD bus, and a SPI bus, a serial bus with a universal asynchronous receiver-transmitter (UART), a Peripheral Component Interconnect (PCI) bus, or a Peripheral Component Interconnect Express (PCIe) bus. Other possibilities exist as well for a device 101 to use a bus 101d to connect electrical components without departing from the scope of the present disclosure.

Device 101 may include a read-only memory (ROM) 101c which can contain a boot loader program. Although ROM 101c is illustrated as "read-only memory", ROM 101c could comprise long-term memory storage chipsets or physical units that are designed primarily for writing once and reading many times, such as Electrically Erasable Programmable Read-Only Memory (EEPROM). As contemplated within the present disclosure, a read-only address could comprise an address within a read only memory such as a ROM 101c memory address or

another hardware address for read-only operations accessible via bus 101d. Changing data recorded in a ROM 101c can require a technician have physical access to device 101, such as removing a cover or part of an enclosure, where the technician can subsequently connect equipment to a circuit board in device 101, including replacing ROM 101c. ROM 101c could
5 also comprise a nonvolatile memory, such that data is stored within ROM 101c even if no electrical power is provided to ROM 101c. In exemplary embodiments, ROM 101c could be (i) optionally omitted and storage radio unit 109 used for nonvolatile memory within device 101, or (ii) included in a memory unit within CPU 101b (not shown).

Device 101 can include a storage radio unit 109, and storage radio unit 109 can be
10 removable. Storage radio unit 109 is also depicted and described in connection with Figure 1a above and additional details regarding storage radio unit 109 are depicted and described in connection with Figures 1c and Figure 2a below. Additional details for the operation of storage radio unit 109 are provided in subsequent figures. An exemplary storage radio unit 109 form factor could comprise a standard SD card, a mini SD card, a micro SD card, a mini UICC, a
15 micro UICC, or a nano UICC, and other possibilities exist as well without departing from the scope of the present disclosure. Storage radio unit 109 could also comprise a solid state hard drive with a suitable small form factor for operation with device 101. Storage radio unit 109 can include electrical contacts (such as external pins 109x depicted in Figure 1c) which provide electrical connectivity to bus 101d.

Storage radio unit 109 can be inserted in a housing in device 101, such that storage radio unit 109 can be manually inserted into or manually removed from device 101 after device 101 is manufactured. The manual operation of insertion of storage radio unit 109 can be performed by entities in some embodiments, such as device service provider 110 or an end user of device 101. Storage radio unit 109 can include NAND or NOR flash memory in order to record data when
25 device 101 is not powered, and other nonvolatile memory technologies can be used in a storage radio unit 109 as well without departing from the scope of the present disclosure. Storage radio unit 109 can be separately manufactured from device 101 and accessed and loaded with data by a configuration unit 110d before insertion into device 101, when storage radio unit 109 includes support for operating as removable media. Storage radio unit 109 could also operate as an
30 “embedded” unit, such that storage unit comprises an integrated circuit soldered to a circuit board in device 101, and in these embodiments storage radio unit 109 can be fixed and not removable.

In exemplary embodiments, storage radio unit 109 can include a radio unit 109b and an eUICC 113 (both not shown in Figure 1b, but shown in Figure 1c), and additional details

regarding the components and operation of an eUICC 113 and radio unit 109b are depicted and described in additional figures below, including Figures 1c and Figure 2a. The inclusion of (i) a radio unit 109b supporting wireless WAN networks 102 and (ii) an eUICC 113, plus the operation of radio unit 109b with eUICC 113 in storage radio unit 109 can add functionality for storage unit 109 that is not normally included in commercially available removable storage media in the market as of 2018, such as SD cards or UICC cards.

Both radio unit 109b and eUICC 113 within storage unit 109 can include a processor, bus, and memory similar (but with less power and on a smaller scale) as the CPU 101b, bus 101d, and RAM 101e and ROM 101c. Both eUICC 113 and SRU 109 can perform cryptographic functions such as (i) internally deriving a private key, recording a private key, or deriving a symmetric key via an elliptic curve Diffie Hellman (ECDH) key exchange, (ii) recording the private key in a protected memory such that device 101 or external parties cannot feasibly or cost-effectively read the private key or derived symmetric key, and (iii) generating or verifying digital signatures using a recorded private key.

Although the exemplary environment described herein employs ROM 101c, RAM 101e, and storage radio unit 109, it should be appreciated by those skilled in the art that storage radio unit 109 could also comprise other types of computer readable media which can store data that is accessible by a device 101, such as memory cards, subscriber identity device (SIM) cards, local miniaturized hard disks, and the like, may also be used in the exemplary operating environment without departing from the scope of the disclosure. The memory and associated hardware illustrated in Figure 1b provide nonvolatile storage of computer-executable instructions, data structures, program devices, device program 101i, and other data for computer or device 101.

Note the device 101 may include a physical data connection at the physical interface 101a such as a miniaturized universal serial bus adapter, firewire, optical, or other another port and the computer executable instructions such as device program 101i, data reporting steps 101x, operating system 101h, or device driver 101g can be initially loaded into memory such as ROM 101c or storage radio unit 109 through the physical interface 101a before device 101 is given to an end user, shipped by a manufacturer to a distribution channel, or installed by a technician. Further, device program 101i, data reporting steps 101x, operating system 101h, or device driver 101g can be separately loaded into storage radio unit 109 before or after distribution of device 101, and then storage radio unit 109 can be inserted into device 101.

A number of program devices may be stored in RAM 101e, ROM 101c, or storage radio unit 109, including an operating system 101h, device driver 101g, an http client (not shown), a DNS client, and related software. Storage radio unit 109 can record program devices as well

within boot firmware, boot configuration, or operating software for storage radio unit 109, which is further depicted and described in connection with Figure 1c and Figure 2a, where the program devices in SRU 109 may be focused on storage within a nonvolatile memory 109a and a radio unit 109b. Program devices include routines, sub-routines, programs, objects,
5 components, data structures, etc., which perform particular tasks or implement particular abstract data types. Aspects of the present disclosure may be implemented in the form of (i) a device program 101i and/or data reporting libraries 101x which are executed by the device 101 working in conjunction with (ii) firmware on storage radio unit 109 (depicted as SRU firmware 109u in Figure 1c and Figure 2a) to support communications of device 101 with a wireless
10 network 102 and device service provider 110 through SRU 109 with RU 109b.

A user may enter commands and information into device 101 through an optional user interface 101j, such as a keypad, keyboard (possibly miniaturized for a mobile phone form-factor), and a pointing device. Pointing devices may include a trackball, an electronic pen, or a touch screen. A user interface 101j may also include a display such as a device screen. A
15 display may also be connected to system bus 101d via an interface. The display can comprise any type of display devices such as a liquid crystal display (LCD), a plasma display, and an organic light-emitting diode (OLED) display. Device 101 may also include a camera (not shown) connected to or integrated with device 101 through a physical interface 101a, and the camera can comprise a video camera for the device 101 to collect sensor data that includes video
20 or images. The camera (not shown) can be a CCD (charge-coupled device) camera, a CMOS (complementary metal-oxide-semiconductor) camera, or a similar device to collect video input. Other arrangements could be used as well, without departing from the disclosure.

The device 101, comprising a computer, may operate in a networked environment using logical connections to one or more remote computers, such as the servers 108 and 110a
25 illustrated in Figure 1a. Servers 108 and 110a can also function as general purpose servers to provide files, programs, disk storage, remote memory, database records, cryptographic data, and other resources to device 101 usually through a networked connection such as via wireless network 102 and IP network 111. Additional remote computers with which device 101 communicates may include another device 101 or mobile device, an M2M node within a
30 capillary network, a personal computer, other servers, a client, a router, a network PC, a peer device, a wireless network 102, an access point, or other common network node. A server 108 or 110a or a remote computer typically includes many of the elements described above relative to the device 101, including a CPU, memory, and physical interfaces. It will be appreciated that the network connections shown throughout the present disclosure are exemplary and other

means of establishing a wireless or wired communications link may be used between mobile devices, computers, servers, corresponding nodes, and similar computers.

The device program 101i and data reporting libraries 101x operating within device 101 illustrated in Figure 1b can provide computer executable instructions to hardware such as CPU 101b through a system bus 101d in order for a device 101 to (i) transmit and receive data with a device service provider 110, (ii) monitor a sensor 101f and/or change the state of an actuator 101y, (iii) send or receive data with a server B 110a, and (iv) authenticate with a server B 110a or SRU service provider 106, thus allowing server B 110a to remotely monitor or control a monitored unit 105 in an authenticated and secure manner. The device program 101i and/or data reporting steps 101x can enable the device 101 to authenticate and communicate with a server B 110b (via SRU 109 and server A 108) by recording data in memory such as RAM 101e, where the data can include sensor data, a destination IP address number such as the exemplary IP address number 110e in Figure 1a, a packet or packet header value, an encryption or ciphering algorithm and key, a digital signature and public key, etc. The data recorded in RAM 101e can be subsequently read by the operating system 101h or the device driver 101g.

The operating system 101h or the device driver 101g can write the data to a physical interface 101a comprising electrical pins 109x on SRU 109 using a system bus 101d in order to use a physical interface 101a to send application data 112a to SRU 109 and radio unit 109b and wireless network 102. Alternatively, the device program 101i and/or data reporting steps 101x can write the application data 112a directly to the SRU 109 using the system bus 101d.

The device program 101i and/or data reporting steps 101x, or operating system 101h can include steps to process the data recorded in memory such as logging data, selecting a destination address, or encoding sensor data acquired by (i) a sensor 101f or (ii) through a physical interface 101a such as a thermocouple, shock or vibration sensor, light sensor, or global positioning system (GPS) receiver, etc. The device 101 can use the physical interface 101a such as a radio to transmit or send (i) the application data 112a from a sensor or (ii) a digital signature of device 101 for application data 112a to SRU 109, where SRU 109 includes a radio unit 109b to send the application data to server A 108 via wireless network 102.

Moreover, those skilled in the art will appreciate that the present disclosure may be implemented in other computer system configurations, including hand-held devices, netbooks, portable computers, multiprocessor systems, microprocessor based or programmable consumer electronics, network personal computers, minicomputers, mainframe computers, servers, and the like. The disclosure may also be practiced in distributed computing environments, where tasks are performed by remote processing devices that are linked through a communications network.

In a distributed computing environment, program devices may be located in both local and remote memory storage devices. In addition, the terms “mobile node”, “mobile station”, “mobile device”, “M2M device”, “M2M device”, “networked sensor”, or “industrial controller” can be used to refer to device 101 as contemplated herein.

5

Figure 1c

Figure 1c is a graphical illustration of hardware, firmware, and software components for a device and a storage radio unit, in accordance with exemplary embodiments. The illustrated exemplary components for a device 101 in Figure 1c can include a device identity 101a, a device
10 clock 101m, a battery 101n, a WiFi radio 101z, a device antenna 101t, a device service provider certificate 110c, and storage radio unit 109. Device identity 101a can be a unique string or numeric identifier for device 101, such as an international mobile equipment identity (IMEI), a MAC (media access control) address, a manufactured serial number, or similar identifier. Battery 101n for device 101 can store electrical energy such that device 101 can be powered
15 when disconnected from an external power source. Clock 101m can be a clock source such as a TCXO or VCTXO to provide a timing source for (i) CPU 101b and bus 101d as well as (ii) SD card driver 101k or other device drivers 101k.

WiFi radio 101z can be a radio module supporting wireless local area network (LAN) connections via standards such as 802.11 from the IEEE. Device antenna 101t can transmit and
20 receive radio frequency signals with other nodes or devices, such as a WiFi router or other devices via WiFi direct. In exemplary embodiments, device antenna supports ultra-high frequency signals in the range of 400 MHz to 6 GHz. Device antenna 101t can have impedance that matches a radio such as 50 ohms. In some exemplary embodiments, device antenna 101t is connected to SRU 109 via electrical pins 109x, such as pins 1 and 8 as shown in Figure 2a,
25 where pins 1 and 8 are not used the 1-bit SD bus mode. Or SRU 109 could include electrical pins or connectors (separate from electrical pin 109x for an SD bus), such as with an external antenna interface 109z depicted and described in connection with Figure 2a. Although the term “SD bus” is used in the present disclosure, other bus technologies can be used for SRU 109 could include versions of a UHS bus, an SD express bus, as well as future, related versions of
30 these bus technologies for removable storage media. Further, although the term “SD card” or “micro SD card” is used for some embodiments of the present disclosure, related card technologies could be use such as cards supporting SDHC, SDXC, SDUC, and subsequent or related versions.

The illustrated components for a storage unit 109 in Figure 1c can include external pins 109x, an SD interface driver 109y, a radio unit 109b, an internal antenna 109i, a storage radio unit identity (SRU-ID) 109i, a power control unit 109e, a memory/interface processor 109d, an eUICC 113, a memory core interface 109m, an interface to an external antenna 109z, an
5 nonvolatile memory 109a, a memory controller 115, and a symmetric key 127a. The external pins 109x of storage radio unit 109 can provide an electrical interface for storage radio unit 109 to connect with a bus 101d of device 101, such that a CPU 101b of device 101 can communicate with the internal elements of storage radio unit 109, using SD card driver 101k.

Note that the use of an SD card and SD bus is not required for storage radio unit 109, and
10 other formats are possible as well, such as storage radio unit 109 comprising an integrated circuit with a PCIe bus or a SPI bus, etc. As mentioned two paragraphs above, other bus technologies for SRU 109 include a version of UHS and SD Express. An SD card can include SDXC, SDUC, SDHC, or future, related versions, and other possibilities exist as well. For the embodiments which do not use SD card formats, then “SD card driver 101k” and “SD interface
15 driver 109y” could comprise drivers appropriate for the physical form of storage radio unit 109 (such as possibly using a PCIe interface with a PCIe driver or similar). The external pins 109x can include pads or electrical contacts for electrical input into storage radio unit 109 and such as a power voltage, electrical ground, a clock input, and data lines to communicate binary data. The external pins 109x can support electrical output from storage unit 109 as well. Exemplary
20 details for pins 109x are depicted and described in connection with Figure 2a below.

In exemplary embodiments, the interface to an external antenna 109z can comprise a female receptacle for the SD card on the opposite side of the electrical pins 109x. The interface to an external antenna 109z can receive 2x male pin connectors preferably less than 0.6 mm in diameter (in order to support the height of a micro SD card), and also a pitch between the center
25 of the 2 pins of less than 2.5 mm in order to reduce space used inside the SRU 109 housing. The length of the male pins can be less than 5 mm in exemplary embodiments. Or, in other exemplary embodiments, the interface to an external antenna 109z can comprise a standard connector on the outside of the SD card housing 227 (depicted in Figure 2a below), such as the male receptacle for an u.fl antenna connection. The interface of the external antenna 109z can
30 comprise a connector both (i) mounted on the opposite side/face as pins 109x and also (ii) on the other end of a housing 227 (shown in Figure 2a below) as pins 109x.

In an exemplary embodiment, the external pins of storage unit 109 could support an SD bus or a serial peripheral interface (SPI) bus mode with a micro SD card, such that pin 6 would be ground, pin 4 would be power, pin 3 would be serial data in, pin 7 would be serial data out,

pin 5 would be a clock signal, etc. Other possibilities exist as well for the possible pin layouts and bus supported by the external pins 109x of an SD card, such as a one-bit SD bus mode or a four-bit SD bus mode. If storage unit 109 comprises a UICC, then the PIN layout of external pins 109x could comply with 3GPP standards and the interface, such as standard 3GPP TS 31.102. In another embodiment, external pins 109x can support either (i) a parallel bus such as a member of the Peripheral Component Interconnect (PCI) standards or a Peripheral Component Interconnect Express (PCIe) standard, or (ii) other serial interfaces such as a universal serial bus (USB).

Interface driver 109y within storage unit 109 can provide or be associated with the firmware to support the selected physical bus type and communications protocol for the external pins 109x. The interface driver 109y can provide functionality similar to device driver 101k or 101g in Figure 1b. For example, interface driver 109y could support one-bit or four-bit SD bus mode for data transfer between (i) the memory/interface processor 109d in SRU 109 and (ii) CPU 101b in device 101, and other possibilities exist as well. The memory/interface processor 109d can comprise a processor similar to CPU 101b, but embedded into the storage radio unit 109 and consequently potentially with less resources than a CPU 101b, such as (i) less cache memory, (ii) operating potentially at a slower speed or clock rate, (iii) fewer internal registers, (iv) lower power consumption due to reduced thermal dissipation for a storage radio unit 109 compared to device 101, etc. Memory/interface processor 109d can also manage and sequence the flow of data between non-volatile memory 109a and the external pins 109x, among other functions such as receiving updated data for radio unit 109b through external pins 109x.

As of 2018, and for the foreseeable future, large capacity removable storage media, such as SD cards or micro SD cards with a gigabyte of memory or more, can include non-volatile memory such as non-volatile memory 109a. Properly managing reading and writing to the non-volatile memory 109a can require relatively computationally intensive operations to be performed inside the removable storage media such as an SD card or storage radio unit 109. These operations performed by memory/interface processor 109d include (i) error correcting codes to reduce or eliminate bit errors in nonvolatile memory 109a, (ii) leveling write wearing in order to spread the number of data writes as uniformly as possible across the physical memory 109a, (iii) registers and tables to keep track of memory, block and page addresses in order to perform operations such as reads and writes to a memory core interface 109m, etc.

These operations can be relatively computationally intensive for a small form factor storage unit such as a micro SD card and can comprise thousands of machine code instructions for a single operation of (i) selecting a page of physical memory within a block of memory, (ii)

reading the selected page of physical memory, (iii) error correcting the data read and also write the error corrections back to the physical memory, and (iv) transferring the data through electrical pins 109x via interface driver 109y. The above exemplary operations be performed by a processor or multiple processors embedded in the storage unit 109 in the form of a memory/interface processor 109d. Exemplary microcontrollers embedded in an SRU 109 include modified versions of an 8051 processor or an ARM® processor, such as an exemplary processor that is a member of the ARM® Cortex M family of processors, and other possibilities exist as well without departing from the scope of the present disclosure. As contemplated herein, memory/interface processor 109d can be referred to as “processor 109d”, “controller 109d”, or “memory controller 109d” or “interface controller 109d”.

Secret symmetric key 127a within SRU 109 can be a secret key for use with symmetric ciphering algorithms. In exemplary embodiments, secret symmetric key 127a can be uniquely associated with SRU 109 using SRU-ID 109i and can also be loaded in storage radio unit 109 either (i) upon manufacturing of storage radio unit 109, or (ii) before distribution of storage radio unit 109 such that symmetric key 127a cannot be subsequently changed or altered after symmetric key 127a is recorded in storage unit 109. A SRU 109 can record multiple secret symmetric keys 127a, such (A) that a first key 127a remains permanently recorded in SRU 109 before installation of SRU 109 with a device 101, while (B) subsequent keys 127a could be recorded in SRU 109 after distribution of storage radio unit 109. As one example, SRU 109 could conduct an ECDH key exchange with SRU-SP 106 using (i) a private key from SRU PKI keys 109p and (ii) a public key from SRU-SP such as public key 106b corresponding to key 106a in order to derive a subsequent, second key 127a for use with encrypting and decrypting application data with SRU-SP 106. In other words, these subsequent symmetric keys 127a could be changed or altered after the first symmetric key 127a was written before distribution of storage unit 109.

Symmetric key 127a or 127b in Figure 1c could comprise a pseudo random number or random number of sufficient length such as an exemplary 128 or 256 bits, such that output of a symmetric ciphering algorithm such as the Advanced Encryption Standard (AES) could not feasibly be decrypted without the use of a symmetric key 127a or 127b. Among many possible uses for symmetric key 127a in the present disclosure, symmetric key 127a could be utilized to decrypt SRU unit firmware 109u, where symmetric key 127a was used to encrypt the SRU firmware 109u before SRU firmware 109u was loaded into storage unit 109. In this manner, SRU unit firmware 109u can be resistant to reading or tampering by end users or other external parties other than the manufacturer of SRU 109 or authorized parties who also have symmetric

key 127a. A symmetric key 127a could also be derived from the network access credentials, such as a key K associated with an IMSI or SUPI, in order to encrypt application data 112a transferred to Server A 108.

Storage radio unit 109 can also include a memory core interface 109m, where memory core interface 109m provides processor 109d and/or radio unit 109b access to physical memory such as non-volatile memory 101a and/or RAM 213 (not shown in Figure 1c but depicted and described in connection with Figure 2a below). Controller 109d and/or radio unit 109b can read data from physical memory by writing an address of a memory page and/or block to memory core interface 109m, and memory core interface 109m returns the data from the page in physical nonvolatile memory 109a or RAM 213.

Similarly, memory processor 109d and/or radio unit 109b can write data from physical memory by writing an address of a memory block and/or page to memory core interface 109m plus the data to be recorded in physical memory, and memory core interface 109m can subsequently write the data to physical memory at the block and page address specified. Other possibilities exist as well for the use of a memory core interface 109m without departing from the scope of the present disclosure. In exemplary embodiments, individual cells within a page can be addressed by a memory core interface 109m as well, and other possibilities for the structure and naming conventions of memory are possible without departing from the scope of the present disclosure.

Storage radio unit 109 can include radio unit 109b, where radio unit 109b was also depicted and described in connection with Figure 1a above, and additional details regarding radio unit 109b are depicted and described in connection with Figure 2a below. Radio unit 109b can include radio processor 109c for controlling radio unit 109b, where processor 109b can be similar to (a) CPU 101b for device 101 from Figure 1b, or (b) memory/interface processor 109d described above. Radio processor 109c can be referred to herein as “processor 109c”. The processor 109c in radio unit 109b can function similar to processor 109d for SRU 109 as described above, with a form factor, speed, and computational capacity suitable for radio unit 109b. Processor 109c in radio unit 109b could be a processor belonging to the exemplary MIPS®, ARM®, or Intel® family of processors, and other possibilities exist as well.

Symmetric key 127b can be equivalent to symmetric key 127a for SRU 109 described above. In exemplary embodiments, a first symmetric key 127b can be used with a TLS-PSK session with SM-SR 107, and in this embodiment the first symmetric key 127b can be recorded in eUICC 113. In addition, a second symmetric key 127b can be used with a DTLS-PSK session with server A 108, where application data 112a is encrypted with the second symmetric key

127g and a symmetric ciphering algorithms, and server B response 112b data is decrypted with the second symmetric key 127b and the symmetric ciphering algorithm. Thus, a radio unit 109b can also operate with multiple symmetric keys 127b, and the multiple pre-shared secret symmetric keys 127b can also include or each be associated with a MAC key, where a MAC key is depicted and described in connection with Figure 4b below. A symmetric key 127b can also separately be used to encrypt and decrypt data for radio unit 109b which is stored in memory 109r.

Processor 109c can include components such as registers, accumulators, and logic elements to add, subtract, multiply, and divide numerical values, and processor 109c can be connected to an internal bus 212 for SRU 109 depicted in Figure 2a below. The timing of processor 109c by an internal clock for SRU 109 as RF clock 205 depicted in Figure 2a below. Note that in some exemplary embodiments, the timing of processor 109c is not driven by device clock 101m (which can be less precise than requirements for NB-IoT communications 104 in Figure 1a), and rather processor 109c is driven by a separate, internal clock 205 within SRU 109. Processor 109c can provide (i) the hardware for radio unit 109b to perform calculations for radio communications with wireless network 102 in addition to (ii) the general operation of radio unit 109b and managing communication between radio unit 109b and other components within SRU 109 via an internal bus 212 of SRU 109 (shown in Figure 2a below). Although processor 109c is depicted in Figure 1c as internal to radio unit 109b, processor 109c could be external to radio unit 109b. Further, in some exemplary embodiments, processor 109d and processor 109c could be combined into a single processor. For these embodiments, the combined processor could have the functionality of the separate processors 109d and 109c depicted in Figure 1c.

The storage radio unit identity (SRU-ID) 109i can comprise a globally unique identifier for SRU 109, such that nodes communicating with device 101 using storage radio unit 109 can properly identify SRU 109 using the SRU-ID 109i. SRU ID 109i can be a string or a number, such as a hexadecimal number with sufficient length such as an exemplary 8 bytes of data for the SRU-ID 109i, although other possibilities exist as well without departing from the scope of the present disclosure. In exemplary embodiments, SRU-ID 109i can be written to hardware such as an EEPROM in SRU 109 upon manufacturing of SRU 109 or configuration by an SRU-SP 106 and SRU-ID 109i is not normally modified or changed after distribution of SRU 109 from an SRU-SP 106 or receipt of SRU 109 by a user. Other possibilities exist as well for the format, recording location, and use of a SRU-ID 109i without departing from the scope of the present disclosure. In exemplary embodiments, SRU-ID 109i can be written to protected

memory or a hardware address within SRU 109 such that a device 101 could not feasibly modify or change the SRU-ID 109i.

eUICC 113 in SRU 109 can comprise an embedded universal integrated circuit card supporting standards such as GSMA SGP.02 or GSMA SGP.22, including subsequent or related versions of these standards. For embodiments where eUICC 113 supports standard SGP.22, then device 101 could operate a local profile assistant (LPA) as an application in device programs 101i, and the eUICC interface family ES10 from SGP.22 could be passed to eUICC 113 via electrical pins 109x. eUICC 113 was also depicted and described above in connection with Figure 1a. eUICC 113 can include an eUICC ID 113a in order to unique identify eUICC 113 across a plurality of globally distributed eUICCs. Although not depicted in Figure 1c, eUICC 113 can record a set of ECC PKI keys for both eUICC and at least one certificate authority. In exemplary embodiments, eUICC 113 records a root certificate for the GSMA for eUICC operations, which eUICC 113 can use to authenticate an SM-DP. eUICC 113 includes a set of cryptographic algorithms for conducting ECDH key exchanges, symmetric ciphering, and digital signature algorithms using elliptic curve cryptography. eUICC 113 can establish a TLS session with SM-SR 107 using a pre-shared secret key (PSK), which could comprise a symmetric ciphering key 127b recorded within eUICC 113 and also be associated with SM-SR configuration data 113b.

As depicted in Figure 1c, eUICC 113 also records SM-SR configuration data 113b, which could include an IP address, domain name, and domain name service (DNS) server IP address in order for the eUICC 113 to resolve a name for SM-SR 107 into an IP address equivalent to IP address 110e depicted in Figure 1a. SM-SR configuration data 113b could also include a PSK for use with a TLS session with SM-SR 107, where the PSK can be a symmetric ciphering key 127b and an associated MAC key (where a MAC key is depicted and described in connection with Figure 4b). eUICC 113 can communicate internally with radio unit 109b using an internal bus 212 as depicted and described in connection with Figure 2a below.

eUICC 113 can record a bootstrap profile 113c, which can include information equivalent to that recorded in a traditional SIM card for connection to a first wireless network 102. eUICC 113 can record an operational profile 113d, which can include information equivalent to that recorded in a traditional SIM card for connection to a second wireless network 102. Or, the bootstrap profile 113c could be used by RU 109b to connect with the second wireless network 102 in a “roaming” mode, and the operational profile 113c could be used by RU 109b to connect with the second wireless network 102 in a “native” mode or with credentials or key K also recorded by the second wireless network 102. The bootstrap profile

113c can be used in a startup or “initial” or “failover” state for eUICC 113 and radio unit 109b, such that connectivity to wireless network 102 could be obtained if a working operational profile 113d is not available for eUICC 113. Operational profile 113d can be used by eUICC 113 and radio unit 109b for preferred operation of device 101 within a wireless network 102.

5 Although illustrated as a separate unit from radio unit 109b in Figure 1c, in exemplary embodiments, eUICC 113 could also be optionally combined with radio unit 109b or processor 109c (where processor 109c could also be combined with processor 109d). In other words, a single physical processor similar to CPU 101b could perform the function of radio processor 109c and eUICC 113. eUICC 113 could comprise also a trusted execution environment (TEE),
10 “secure processing environment”, “smart secure platform” or similar secured environment within a larger, more generalized processor such as processor 109c.

Non-volatile memory 109a in storage radio unit 109 can include device and SRU accessible nonvolatile memory 109f and SRU only accessible nonvolatile memory 109r. The two types of memory can be identified by separate addresses within the physical memory
15 comprising non-volatile memory 109f, or similarly different sectors assigned by a file system written to the physical nonvolatile memory. Further, memory 109f and 109r could represent different partitions within a memory 109a, where memory 109a has been formatted with a file system. In an exemplary embodiment the two types of memory 109f and 109r can be segmented logically, where memory 109r is encrypted with a symmetric key 127a within SRU 109 by a
20 memory controller 115, where memory controller 115 is also depicted and described in connection with Figure 2a below.

Device and SRU accessible memory 109f can include shared nonvolatile memory 109h and device general memory 109g. Device general memory 109g can comprise memory that device 101 can access in non-volatile memory 109a such as recording firmware for device 101
25 or other long-term and non-volatile storage of data or files for device 101. In exemplary embodiments, operating system 101h for device 101 can be recorded in device general memory 109g. Device and SRU accessible memory 109f can record a file system for both device 101 and SRU 109 to access, such as exemplary file systems of FAT16, FAT 32, NTFS, ext3, ext4, UDF, and other file systems for memory 109g and memory 109h are possible as well without
30 departing from the scope of the present disclosure.

As contemplated herein, memory 109f can represent physical memory cells within a nonvolatile memory 109a, and memory 109f and memory 109r can represent separate file systems or logical and secure separation of memory 109f and 109r. In exemplary embodiments, a file system within memory 109f can be specified as compatible with a file system for OS 101h

for device 101, while a file system within memory 109r can be a different file system and file system type preferred for internal use by SRU 109. For an exemplary embodiment, the file system type for memory 109f can comprise a FAT32 or exFAT file system, and memory 109r can comprise a JFFS2 or ext2 file system. Other possibilities exist as well regarding the use of a file system for memory 109f and memory 109r, without departing from the scope of the present disclosure. Additional details regarding the format, structure, and files within nonvolatile memory 109a and 109h will be provided below in connection with Figures 2a, 2b, 2e, and 2f, as well as other descriptions herein.

Shared memory 109h in nonvolatile memory 109f can include memory accessible by both a device 101 and SRU 109, such as a portion of a file system recorded in memory 109f. Exemplary use of device and SRU shared memory 109h is depicted and described in connection with Figure 2a below. In exemplary embodiments, both device 101 and SRU 109 can both read and write to device and SRU shared nonvolatile memory 109h, which can also be referenced herein as shared nonvolatile memory 109h or shared memory 109h. As illustrated in Figure 1c, shared memory 109h can be a subset of device and SRU accessible memory 109f, and device and SRU accessible memory 109f can also include general memory 109g for device 101. Note that (i) RU 109b can read and write directly to shared memory 109h, or (ii) RU 109b can read and write to shared memory 109h through a memory processor 109d in SRU 109. In other words, for (ii) RU 109b could communicate to memory processor 109d via an internal bus, and processor 109d could read and write to memory 109h on behalf of RU 109b.

In exemplary embodiments, the SRU 109 may have access to general memory 101g, such as to support commands or instructions from SRU-SP 106, or device 101, or device service provider 110, which could comprise instructions to fix memory, reallocate file space or structure, or other system administration operations. However, without outside instruction to SRU 109 from device 101 or SRU-SP 106 or device service provider 110, in regular operating modes SRU 109 may preferably not use memory 109g for the internal operation of SRU 109, where memory 109g can comprise memory and files for general nonvolatile storage of data for device 101. In exemplary embodiments, memory 109h can comprise a first directory within a file system for memory 109f and memory 109g for device 101 can comprise a second directory within a file system for memory 109f. Other possibilities for the configuration of a non-volatile memory 109a are possible as well, such that storage radio unit 109 can provide nonvolatile memory (i) dedicated to device 101, (ii) dedicated to SRU 109, and (iii) shared between device 101 and SRU 109 for passing data between device 101 and RU 109b, without departing from the scope of the present disclosure.

Data utilized by radio unit 109b may logically or physically separated for use by radio unit 109b and processor 109c or processor 109d through the use a memory controller 115. The data used by radio unit 109b and processors 109c and 109d can be recorded in SRU only accessible nonvolatile memory 109r. As contemplated herein, SRU only accessible nonvolatile memory 109r can be also be referred to as “SRU memory 109r”, “SRU 109 protected memory 109r”, “SRU accessible memory 109r”, or “protected memory 109r”.

Memory controller 115 can provide physical or logical isolation of SRU 109 only accessible memory 109r such that device 101 could not read or write to SRU only accessible memory 109r. Memory controller 115 could prevent device 101 from reading or writing data to SRU 109 only accessible memory 109r. As one example, the physical memory in storage unit 109 comprising non-volatile memory 109a could have a portion of the physical addresses reserved for reading and writing by SRU 109 only, such as an exemplary top 50 blocks of non-volatile memory 109a, and other values are possible as well. Memory controller 115 could intentionally disallow the transfer of data from or to non-volatile memory 109r with external bus 202, where the block address is in the exemplary top 50 blocks.

Memory controller 115 could also operate on a lower level than a block address for non-volatile memory 109a as well, such as only allowing SRU 109 or processor 109c and processor 109d to allow a specified range of pages within non-volatile memory 109a, where the specified range of pages could belong to SRU only accessible memory 109r. In this manner, memory controller 115 could operate as a firewall to restrict access to SRU only accessible memory 109r. Other possibilities exist as well for the operation of a memory controller 115 in order to isolate and separate SRU only accessible memory 109r such that interface controller 109d or device 101 cannot utilize physical memory addresses for recording and reading data utilized in a RU only accessible memory 109g.

In another embodiment, memory controller 115 could perform hardware-based encryption/decryption using (i) a symmetric key 127b to encrypt and decrypt data transferred between RU 109b and SRU only accessible memory 109r, or (ii) symmetric key 127a to encrypt and decrypt data transferred between processor 109d and SRU memory 109r. In an exemplary embodiment, SRU only accessible memory 109r can be formatted with a file system with a separate partition from the memory that device 101 accesses within memory 109f. A file system on SRU only accessible memory 109r could be encrypted using a symmetric ciphering algorithm (from cryptographic algorithms 109q) and (i) a symmetric key 127b recorded by RU 109b or (ii) symmetric key 127a used by processor 109d, such that even if RU only accessible memory 109r could be accessed by device 101, no useful data could be extracted or tampered with (since

symmetric keys 127a and 127b could also include MAC keys for data integrity). Although memory controller 115 is depicted in Figure 1c as a separate element from memory core interface 109m, memory controller 115 could also be optionally integrated with memory core interface 109m, and other possibilities exist as well for the operation and location of a memory controller 115 for ciphering data or restricting access to data recorded in storage unit 109 for SRU 109 with protected memory 109r, without departing from the scope of the present disclosure.

As depicted in Figure 1c, SRU memory 109r can include a radio unit AT control stack 109s, SRU firmware 109u, PKI keys 109v, and cryptographic algorithms 109q. Other data for the operation of SRU 109 and radio unit 109b can be recorded in SRU memory 109r as well, such as the exemplary data depicted and described below for protected memory 109r in Figure 2a. AT control stack 109s can comprise software and firmware for SRU 109 and processor 109d to operate as an AT client for radio unit 109b, where processor 109c operates as an AT host for radio unit 109b. As of 2018, most commercially available radio modules for devices such as an RU 109b may be driven or operated with an AT command set.

An SRU 109 could include a commercially available radio unit 109b or similarly a module (possibly as a wafer level chip scale package) in order to fit within a micro SD card or SRU 109 with a housing 227 from Figure 2a below. A commercially available module for RU 109b may require AT command sets such as those specified by 3GPP in TS 27.007. Consequently, in order to utilize a commercially available module for radio unit 109b, SRU 109 could record radio unit AT control stack 109s. For embodiments where AT command controls are not utilized by radio unit 109b, then SRU 109 could record and use the client-side commands and instructions in stack 109s.

SRU firmware 109u can comprise the firmware, software, operating system, and various libraries or packages for SRU 109 to operate after a boot process. Additional details regarding SRU firmware 109u are provided in Figure 2e below. PKI keys 109v can record private keys for use by SRU 109 and RU 109b, such as a SRU private key SRU SK 109p from Figure 1a and public keys for SRU-SP 106 such as SRU-SP PK 106b. PKI keys 109v can include multiple PKI keys, such as a public key for SRU-SP 106 to use with code verification or signatures on SRU firmware 109u. In other words, SRU-SP 106 could provide or record the firmware 109u with a digital signature, and SRU 109 could verify the digital signature using a public key from SRU-SP 106 in memory 109r. PKI keys 109v could also record ephemeral PKI keys derived by SRU 109 or RU 109b. PKI keys 109v could also record a public key for SRU 109 for use by device 101 or SRU-SP 106 or device service provider 110, such as SRU public key SRU PK

109w, and SRU PK 109w could be recorded with a certificate for SRU 109. PKI keys 109v can also record a set of cryptographic parameters 191e or cryptographic settings 192e (depicted in Figure 2b below and later figures as well, such as Figure 4b).

In exemplary embodiments, initial PKI keys for SRU 109 (such as keys 109p and 106b, as well as associated certificates and certificate issuer root certificates) may be recorded in EEPROM as shown in Figure 2a, and subsequent or updated versions of the keys can be recorded in memory 109r. PKI keys 109v can also record a set of root certificates for SRU 109 to use with verification of certificate authority digital signatures for certificates such as certificate 110c. Root certificates recorded in PKI keys 109v can comprise the list or a subset of the list of included root certificates from the Mozilla Foundation with Mozilla projects, where the aggregate list of community approved root certificates and associated parameters is in the widely distributed file certdata.txt from the Mozilla web site.

Cryptographic algorithms 109q in memory 109r can comprise digital signature algorithms such as ECDSA or DSA, symmetric ciphering algorithms such as AES and TripleDES, key exchange algorithms, key pair generation algorithms, random number generators, secure hash algorithms such as SHA-2 or SHA-3, and asymmetric ciphering algorithms such as El gamal or RSA. Libraries in cryptographic algorithms 109q could be from standard open source libraries such as OpenSSL, crypto++, Bouncy Castle, or Mozilla libraries from Network Security Services, and other possibilities exist as well.

Figure 2a

Figure 2a is a graphical illustration of components within a storage radio unit, in accordance with exemplary embodiments. Figure 2a depicts additional details for a radio unit 109b and storage radio unit 109 illustrated in Figure 1c. A system 200a can comprise a SRU 109 and an external antenna 201i. Note that external antenna 201i could be within device 101 and could also be the same as a device antenna 101t. Storage radio unit 109 can include nonvolatile storage memory 109a, radio unit 109b, eUICC 113, phased loop logic (PLL) 211, memory core interface 109m, memory/interface processor 109d, radio processor 109c, external data bus 202, an internal bus 212, electrical pins 109x, power detector 209, power capacitor 208, RAM memory 213, memory controller 115, EEPROM 216, internal antenna 109t, interface to external antenna 109z, and SD driver 109y, and power control unit 109e. Storage radio unit 109 can be enclosed in an SRU housing 227, where SRU housing 227 can be compatible with standard physical housings such as a standard SD card, a micro SD card, a mini UICC, or possibly an integrated circuit package. Figure 2a also depicts an external antenna 201i which

can be connected to the interface to external antenna 109z. Although radio processor 109c is depicted external to radio unit 109b in Figure 2a, radio processor 109c could also comprise a logical or physical component in radio unit 109b. Or, the function of radio processor 109c and memory/interface processor 109d could be combined into a single processor.

5 Electrical pins 109x are depicted as the pins for a micro SD card supporting 1 bit SD bus mode, although other possibilities exist as well for the physical electrical connections for SRU 109. Electrical pins 109x can comprise pin 1 201a and pin 8 201b for connection to a device antenna such as device antenna 101t in Figure 1c, pin 2 201c for chip select (CS), pin 3 201d for data input (MOSI) to SRU 109, pin 5 201e clock input from an clock in device 101 such as
10 clock 101m, pin 6 201f for data output (MISO) from SRU 109, pin 7 201g for electrical ground, and pin 4 201h for powering voltage input (which ranges from 2.7 to 3.6 volts for the micro SD card format).

 Electrical pins 109x can be connected to an external bus 202, which can be connected to memory/interface processor 109d. Electrical pins 109x do not need to be electrically combined
15 into a single circuit, and the separate pins, traces, or wiring lines can be separately connected to processor 109d, and the external facing pins can collectively be considered an external bus 202. In exemplary embodiments, an SD card format and electrical pin connections are not required, and other physical forms of electrical connections could be supported as well, such as PCIe, SPI, serial, or USBv3. Note that pins 1 201a and pin 8 201b can be used as signaling lines for
20 various SD bus technologies, and not with a device antenna for some embodiments, such as embodiments with the use of an external antenna 201i.

 As depicted in Figure 2a, radio unit 109b can include an RF front end 203, an RF transceiver 204, an RF clock 205, a baseband unit 206, a baseband clock 207, and an eUICC 113. Radio unit 109b can be connected to internal bus 212 in order to send and receive electrical
25 signals with both radio processor 109c and memory/interface processor 109d, as well as the other electrical components connected to internal bus 212. As noted above, RU processor 109c could also operate within RU 109b. RF front end 203 can include SAW filters, a power amplifier for a transmit chain, a low noise amplifier for a receive chain, a switching circuit for to switch between transmit and receive in half-duplex mode, and logic and circuits to convert radio
30 frequency signals between an antenna and an intermediate frequency for processing by the RF transceiver 204. RF front end 203 can include an oscillator, phased loop logic, and a mixer and supporting components.

 In exemplary embodiments, RF front end 203 includes an antenna switch 203a. Antenna switch 203a can provide connectivity to each of device antenna 101t through pin 1 201a and pin

8 201b, the interface to external antenna 109z, and internal antenna 109t. In an exemplary embodiment, switch 203a can be selected or set to use one of the three antennas by radio processor 109c using general purpose input/output (GPIO) lines, or similar signals from radio processor 109c. In another exemplary embodiment, switch 203a can automatically determine
5 which of the three antennas (101t in device 101, 109t, or 201i) to use with RF front end 203 based on detecting a 50 ohm impedance along the three possible connections (e.g. pins 1,8 of 109x, through 109z, or with 109t). In exemplary embodiments where RU 109b detects a matching impedance for at least two of the exemplary three antennas depicted (e.g. 101t in device 101 in Figure 1a, 109t, or 201i), then RU 109b could use multiple in and multiple out
10 (MIMO) radio modulation.

RF transceiver 204 can include electrical components to convert between (a) intermediate frequencies with RF front end 203 and (b) baseband frequencies and signals with baseband unit 206. RF transceiver 204 can include digital to analog converters, analog to digital converters, and also use input from RF clock 205 for timing signals. RF transceiver 204 can
15 implement (i) OFDM and QPSK signal modulation and (i) separate RF channels with multiple sub-carriers required by wireless network 102, if wireless network supports LTE and 5G and related standards. High order modulation could be supported such as 16-QAM, 64-QAM, 256-QAM, etc. For low bandwidth applications, BSPK or QPSK could be used.

In exemplary embodiments, RF clock 205 frequency is selected to match or support the
20 wireless WAN networking protocol, such as using a RF clock frequency of 38.4 MHz for LTE NB-IoT applications, although other frequencies could be utilized as well, such as, but not limited to, 19.4 Mhz. Note that SRU 109 may need a separate, internal clock comprising RF clock 205 than clock 101m and input clock via 201e, since that clock source for SD cards or an SD bus or similar PCI or SPI busses is normally not sufficiently accurate or stable for a radio
25 unit 109b. RF transceiver 204 can implement logic and algorithms for an intermediate frequency in order to support the protocol used by wireless network 102, which could be NB-IoT, 5G, and other possibilities exist as well.

Baseband unit 206 can include electrical components to convert between baseband frequency signals communicated with RF transceiver 204 and data with radio processor 109c.
30 Baseband unit 206 can implement radio control functions such as signal modulation, encoding, radio frequency shifting, carrier selection, frequency tuning. For NB-IOT 104 networks, baseband unit 206 can implement baseband physical channels, such as processing narrowband primary synchronization signal (NPSS) and narrowband secondary synchronization signal

(NSSS) to synchronize its frequency and timing with an eNB. Radio unit 109b can adjust or offset clock input from clock 205 in order to synchronize timing with an eNB.

After RU 109b synchronizes timing with wireless network 102, baseband unit 206 by receiving data from RF transceiver 204 can process and read data from a narrowband broadcast control channel (NPBCH). Narrowband physical random access channel (NPRACH) can be implemented by baseband unit 206 to establish initial association with wireless network 102. Baseband unit 206 can structure downlink and uplink data packets with a narrowband physical downlink shared channel (NPDSCH) and narrowband physical uplink shared channel (NPUSCH), respectively. Although baseband unit 206 is depicted as separate from radio processor 109c and RF transceiver 204, baseband unit 206 could be combined with either radio processor 109c and/or RF transceiver 204. In exemplary embodiments, baseband unit 206 can be driven by a baseband clock 207, which can operate at a different and lower frequency than RF clock 205. In some exemplary embodiments, baseband clock 207 operating at 32.768 kHz, although other possibilities exist as well.

As depicted in Figure 2a, eUICC 113 can also be implemented within radio unit 109b, and connected to a processor 109c operating in radio unit 109b via an internal bus for radio unit 109b. Or, eUICC 113 can be implemented as external to radio unit 109b and connected to radio unit 109b and processor 109c via internal bus 212 for SRU 109 (which is depicted in Figure 1c). PLL 211 operating in SRU 109 can comprise a timing circuit to convert RF clock 205 into a frequency to drive processors 109c and/or 109d. In an exemplary embodiment, radio transceiver 204 can be driven by a RF clock 205 with a frequency in a range of less than 50 MHz, such as an exemplary 38.4 or 19.2 MHz, while processors 109c and 109d can preferably be driven at a higher frequency such as a frequency greater than 100 Mhz.

PLL 211 can step up or convert the frequency of RF clock 205 into a higher value or frequency. In an exemplary embodiment where RF clock 205 operates at 38.4 MHz, PLL 211 can multiple the frequency of RF clock 205 to a higher value of 5x times, and thus output a frequency of 192 MHz to drive processor 109c and processor 109d. PLL 211 could use different multiplying RF clock value 205 other than the exemplary value of 5, such as exemplary integer values in the range of 3x to 8x, and other possibilities exist as well. Note that the output of PLL 211 can also be input to eUICC 113, since eUICC 113 can operate with an internal processor in order to perform cryptographic operations. PLL 211 can implement separate multiplication values of RF clock 205 in order to drive each of processor 109c, processor 109d, and eUICC 113 with different clock input values, using RF clock 205 as a source. PLL 211 can function as an

oscillator (i) receiving input from RF clock 205, and (ii) outputting a CMOS timing signal at 100 megahertz or higher.

Random access memory 213 in SRU 109 can function similar to RAM 101e for device 101, with a form factor, speed, and storage capacity suitable for SRU 109. RAM 213 can be connected to internal data bus 212 in storage radio unit 109, and can store data for processors 109c and 109d in a volatile state, such that data recorded in RAM 213 can be essentially flushed or reset when electrical power to storage radio unit 109 and radio unit 109b is removed. Random access memory 213 can store data such as tables of memory addresses and sectors for memory 109f and memory 109r, since these tables are ordinarily larger than the registers provided by processors 109c and 109d. RAM 213 can be shared or allocated between processors 109c and 109d, or RAM 213 can comprise two separate RAM units, where one RAM memory unit is allocated to processor 109c and a second RAM memory unit is allocated to processor 109d.

In addition, for some exemplary embodiments, RAM 213 can comprise a nonvolatile RAM (NVRAM) as described in connection with Figure 5a and depicted for a RAM memory 101e' in Figure 8. Since SRU 109 may be powered off frequently and at times not controlled by SRU 109 (such as under the control of device 101), for some exemplary embodiments RAM 213 comprises an NVRAM. For these embodiments, the time required for a boot process for SRU 109 and RU 109b can be reduced (since a last operating state can be stored in RAM 213 as NVRAM), and thus power or energy consumption by SRU 109 can be reduced.

EEPROM 216 in SRU 109 can include SRU ID 109i, a certificate 106d for SRU-SP 106 with an initial SRU-SP public key SRU-SP PK 106b corresponding to the SRU-SP private key SRU-SP SK 106a, symmetric keys 127, SRU boot firmware 217, SRU boot configuration 218, and PKI keys 109v' for SRU 109. Symmetric keys 127 can comprise a collection of pre-shared secret keys 127a and 127b. EEPROM 216 could also comprise a read only memory, where the data in EEPROM 216 is written once upon manufacturing of storage unit 109. EEPROM 216 could also function as a read only memory for SRU 109 similar to ROM 101c for device 101 above. In other words, EEPROM 216 does not need to be erasable and reprogrammable, although some data in EEPROM 216 could be re-written in exemplary embodiments.

Data within EEPROM 216 can comprise SRU boot firmware 217 and CU boot configuration 218. Upon startup or powering of storage radio unit 109, processor 109d could load and operate on instructions provided by boot firmware 217 and boot configuration 218 from EEPROM 216. Boot firmware 217 and boot configuration 218 can provide instructions to processor 109d in the format of machine executable code. The instructions in boot firmware 217 can provide instructions for processor 109d and possibly processor 109c to initialize and

subsequently load SRU firmware 101u from memory 109r via internal bus 212. Data or files can be written to EEPROM 216 by a configuration unit similar to configuration unit 110d, where the configuration unit for EEPROM 216 can be operated by (i) the manufacturer of SRU 109 or (ii) SRU-SP 106. The data in EEPROM 216 can be written into EEPROM 216 in SRU 109 before SRU 109 is distributed to a device service provider 110, an SRU distributor, or an end user of device 101, such as during a step 260 depicted and described in connection with Figure 2f below.

PKI keys 109v' in EEPROM 216 can comprise an initial set of PKI keys 109v depicted and described in connection with Figure 1c, with a difference being (a) PKI keys 109v' could be recorded in a read-only state upon manufacturing of SRU 109, and (b) PKI keys 109v in nonvolatile memory 109a could comprise a subsequently updated version of the initial set of PKI keys 109v'. PKI keys 109v' can include a certificate for SRU 109. A certificate for SRU 109 in PKI keys 109v' can include the SRU identity 109i, SRU public key SRU PK 109w for SRP private key SRU SK 101p, certificate parameters, and a certificate authority digital signature.

SRU certificate for SRU 109 (and other certificates herein) can be formatted according to the X.509 v3 specifications, among other possible formats, and stored as a plain text file, *.pem file, or *.crt file or similar file formats. SRU certificate in PKI keys 109v' can be used by SRU 109 in order to (i) verify identity of SRU 109 to device 101 or SRU-SP 106. SRU private key 109p in PKI keys 109s can be used to generate a digital signature. In exemplary embodiments, parameters in a SRU certificate in PKI keys can include an expiration time of longer than the expected operational lifetime of storage radio unit 109, and in this manner SRU certificate in PKI keys 109s can remain valid whenever storage unit 109 is utilized. An exemplary expiration time of SRU certificate in PKI keys 109s could be 20 years, although other possibilities exist as well. For embodiments where a certificate for SRU 109 is updated (such as changing a set of cryptographic parameters 191e), then the updated certificate for SRUP 109 could be stored or recorded in PKI keys 109v in protected memory 109r and the original certificate could continue to be recorded in EEPROM 216.

SRU boot firmware 217 in EEPROM 216 can provide machine executable code for processor 109d to initiate operations when electrical power is provided to SRU 109 via the electrical pins 109x. Although not illustrated in Figure 2a, processor 109d may also include a ROM memory with processor 109d instructions for processor 109d to fetch boot firmware 217 and boot configuration 218 upon startup when power is provided to processor 109d. SRU boot firmware 217 can include memory control libraries 217a. SRU boot firmware 217 in EEPROM

216 could also include instructions for SRU 109 to load SRU firmware 109u recorded in non volatile memory 109a, if present, as depicted in Figure 2a.

In exemplary embodiments, SRU firmware 109 could be (i) recorded in memory 109r as encrypted with symmetric key 127a, and (ii) decrypted with a symmetric key 127a recorded in EEPROM 216 within symmetric keys 127. Authorized providers of SRU firmware 109u, such as SRU-SP 106 or device service provider 110, could have access to symmetric key 127a for SRU 109 with SRU ID 109i, and consequently only the authorized providers could properly encrypt SRU firmware 109u using the symmetric key 127a. In exemplary embodiments, symmetric key 127a is unique for each SRU 109 and associated with each unique SRU-ID 109i. SRU firmware 109u could be transmitted to a device 101 over network 102 in the encrypted format and encrypted with the symmetric key 127a for SRU 109.

Or, SRU firmware 109u can be written to SRU 109 by a configuration unit similar to configuration unit 110d and operated by SRU-SP 106 or device service provider 110 or a manufacturer for SRU 109, and other possibilities exist as well for the source and recording of an initial SRP firmware 109u for SRU 109 before SRU 109 is inserted in device 101. Symmetric key 127a within EEPROM 216 can comprise a shared secret key. Symmetric key 127b for SRU 109 can comprise a different key for other purposes than decrypting SRU firmware 109u. A configuration unit 110d could write a symmetric key 127b to EEPROM 216 or protected memory 109r, where the symmetric key 127b could be used to encrypt application data 112a and decrypt a response 112b. In some exemplary embodiments, SRU firmware 109u is not required for SRU 109 to operate, and SRU 109 could operate using SRU boot firmware 217. SRU firmware 109u can be loaded into storage unit 109 for embodiments where the firmware for SRU 109 is desired to be updated.

Memory control libraries 217a could include software or firmware to manage and schedule the operation of SRU 109, such as machine code or machine executable instructions for (i) instructing processor 109d to write data to bus 212 for memory controller 115 when data is recorded in memory 109r (e.g. memory write libraries 217b), (ii) read data from external bus 202 when data from device 101 is passed to SRU 109 (e.g. memory read libraries 217c), and (iii) reading SRU firmware 109u from protected memory 109r for SRU 109. For embodiments where processor 109c and processor 109d are combined, memory control libraries 217a can include the software libraries and firmware for the combined processor to manage all input and output of SRU 109 upon boot. Other possibilities exist as well for memory control libraries 217a to support the operation of SRU 109 via program instructions provided to processor 109d without departing from the scope of the present disclosure. In exemplary embodiments, boot

firmware 217 and boot configuration 218 can provide boot instructions and setup for both processor 109c and processor 109d.

SRU only accessible memory 109r within nonvolatile memory 109a could be accessed by SRU 109 and processors 109d and 109c via an internal bus 212 and memory controller 115.

5 SRU only accessible memory 109r within nonvolatile memory 109a can comprise a protected memory (such that memory 109r cannot be read by device 101 or written to by device 101). As contemplated herein, SRU only accessible memory 109r can also be referred to as protected memory 109r. Internal bus 212 and memory controller 115 can be utilized to physically or logically separate protected memory 109r from memory 109f and memory 109g in Figure 1c, in
10 order to prevent device 101 or external parties with physical access to storage unit 109 from reading the data or writing data in protected memory 109r. In other words, and as illustrated in Figure 2a, internal bus 212 and memory controller 115 can limit (A) the ability to read and write data to protected memory 109r only to (B) SRU 109 in exemplary embodiments. In contrast, access by device 101 to memory 109f could be accomplished through the separate data bus 202
15 and interface controller 109d.

In exemplary embodiments, memory processor 109d can provide the functionality to device 101 for storage radio unit 109 to operate as a standard flash memory storage unit via data bus 202. Internal bus 212 and memory controller 115 can provide memory resources to SRU 109 in a manner that prevents device 101 from accessing protected memory 109r, including the
20 description of memory controller 115 above. Internal bus 212 is not required in some exemplary embodiments, and memory controller 115 could be connected to data bus 202, but limit access to physical memory cells comprising memory 109r to processors 109d and 109c. Other possibilities for (i) limiting access to memory 109r to SRU 109, while (ii) supporting shared access to memory 109h with device 101 are possible as well without departing from the scope of
25 the present disclosure. In an exemplary embodiment, memory 101f, and 109r could represent separate partitions for nonvolatile memory 109a.

SRU only accessible memory 109r in nonvolatile memory 109a can include data for the operation of SRU 109 which would not normally be read or utilized by device 101. In addition to the exemplary data depicted and described for protected memory 109r in Figure 1c, memory
30 109r can include an operating system (OS) 220 for SRU 109, cryptographic algorithms 109q, device service provider certificate 110c, an SRU-SP public key SRU-SP PK 106b, radio unit firmware 222, derived secret shared key 223, free memory 224 for SRU 109, and SRU firmware 109u. Data within memory 109r can be recorded in a file system similar to file system 229 for device memory 109g. OS 220 for SRU 109 can be an OS similar to the OS 101h for device 101,

but selected for the operating environment of SRU 109. Cryptographic algorithms 109q were described above in connection with Figure 1c. Device service provider certificate 110c can be used to verify a digital signature for data in a server B response 112b. SRU-SP public key SRU-SP PK 106b corresponds to SRU-SP private key SRU-SP SK 106a from Figure 1a. Radio unit
5 firmware 222 can comprise software, firmware, and operating libraries for radio unit 109b, including radio unit processor 109c.

Derived secret shared keys 223 can comprise secret shared keys derived between SRU 109 and SRU-SP 106 using an ECDH key exchange and SRU private key SRU SK 109p and SRU-SP public key SRU-SP PK 106b. Derived shared secret keys 223 can also comprise secret
10 shared keys derived with SRU-SP using an ephemeral private key derived by SRU 109 such as ephemeral private key 409a as depicted and described in Figure 4a and Figure 4b below. Free memory 224 for SRU 109 can comprise memory allocated for SRU 109 in protected memory 109r that is available for future use by SRU 109 during operations after a boot process. SRU firmware 109u is described above in Figure 1c and also below in Figure 2e. Other data could be
15 recorded in protected memory 109r as well, including an updated boot configuration 218, which could include boot parameters for the operation of RU 109 that are different than initially stored SRU boot configuration 218 in EEPROM 216.

Power control unit 109e can manage the power utilization by SRU 109. Since many applications of device 101 may operate on a battery 101n, current flow and energy consumption
20 by SRU 109 can be managed by power control unit 109e. Power control unit 109e can also include a voltage regulator in order to limit a maximum input voltage from pins 109x to components in SRU 109 (and can be connected to bus 202). Power control unit 109e can smooth or reduce voltage fluctuations from power received by device 101. In exemplary embodiments, power control unit 109e can either (i) disable power input into radio unit 109b or
25 (ii) force radio unit 109b to enter a sleep state when device 101 writes to memory 109a. The disabled or sleep state for radio unit 109b could comprise current utilization for radio unit 109b of less than an exemplary 5 micro amps. Power control unit 109e can also (i) change radio unit 109b from a booted but sleep state to (ii) an active mode or operating mode (e.g. ready to receive data and transmit data) with operating power for radio unit 109b after data is written by
30 device 101 into shared memory 101h, such as application data 112a from device 101 for server B 110a. In other words, power control unit 109e can keep radio unit 109b in a deep sleep state until SRU receives data from device 101 in shared memory 109h, where the data from device 101 in shared memory 109h is for radio unit 109b to transmit through wireless network 102.

Exemplary additional steps and function for a power control unit 109e are depicted and described in subsequent figures below, including Figure 6a and Figure 6b.

Further, power control unit 109e could disable writes or erases by device 101 into device memory 109g when radio unit 109b is transmitting, since those writes or erases in combination with radio unit 109b transmitting may exceed specified or preferred current consumption thresholds for storage radio unit 109. In exemplary embodiments, radio unit 109b can consume approximately 200 mA of current for several seconds when transmitting at full power of 23 dBm. Similarly, SRU 109a can consume approximately 50 -- 75 mA of current when erasing blocks of memory for device 101 in general memory 109g. Other possibilities exist as well, but those values are typical for radio units and flash memory units commercially available in 2018.

In addition, power control unit 109e could prevent writes or erases of data in device memory 109g when radio unit 109b is transmitting in order to reduce possible internal radio frequency noise. Power control unit 109e could control power input or supply voltage to radio unit 109b, such that input power to radio unit 109b must pass through power control unit 109e. Power control unit 109e could have electrical connections to radio unit 109b to signal sleep and wake signals, possibly through internal bus 212. For some exemplary embodiments, RU 109b can comprise a radio supporting 802.11 ah standards, which are also known as WiFi Halow. The sleep of RU 109b and/or SRU 109 can be controlled by a power control unit 109e, and power control unit 109e can receive a target wake time (TWT) from a wireless network 102. Power control unit 109e can power up or down the radio unit 109b and SRU 109 based on the TWT received. In exemplary embodiments, SRU 109 can write the TWT and a corresponding date/time value to device read memory 192, such that device 101 can read the TWT and plan or schedule to take sensor measurements based on the TWT received from the wireless network 102 by radio unit 109b.

Power control unit 109e could also utilize power detector 209. Power detector 209 could be connected to power control unit 109e via an internal bus 212. Power detector 209 can (i) detect when power or voltage is removed from pin 4 201g (or similar pins for voltage input with other busses) and (ii) send a shutdown signal to power control unit 109e. Power control unit 109e can receive the shutdown signal and conduct a clean shutdown of SRU 109 and radio unit 109b. For embodiments where RAM 213 comprises NVRAM, then a shutdown command from power control unit 109e after (x) power detector 209 determines that power from device 101 has been removed can then (y) instruct SRU 109 to store the last operating state in NVRAM (similar to a "hibernate" shutdown for a personal computer). Thus, SRU 109 can resume operation more quickly once power is restored to SRU 109 using the last operating state from NVRAM. Power

detector 209 and power control unit 109e operating together can remove power from RU 109b within fractions of a millisecond when RU 109b is transmitting and power is removed from SRU 109.

Power capacitor 208 could provide sufficient charge storage to allow SRU 109 to operate for at least a few milliseconds after power supply voltage is removed from pin 4 201g (or similar pins/connectors for voltage input with other busses). In this manner, power control unit 109e can complete outstanding write operations to memory 109f (e.g. memory 109h and memory 109g) or memory 109r, and avoid data being corrupted. In an exemplary embodiment, power capacitor could be a surface mount capacitor supporting approximately 20 - 200 micro farads at the low voltage needed by SRU 109, such as a maximum supporting voltage of 5 volts. SRU could consume an exemplary 25 - 50 mA during a device 101 write operation to memory 109f or radio transmission by RU 109b (such as with 0 – 15 dBm RF transmit power). Voltage supplied to SRU 109 can be an exemplary 3.3 volts.

As an exemplary calculation, if power was suddenly removed from pin 4 201g, then SRU could use charge stored in power capacitor 208 with an exemplary 100 microfarads and a voltage drop of 0.6 volts and a current draw of 33 mA to operate SRU 109 for an additional 2 milliseconds. SRU 109 using all of power detector 209, power capacitor 208, and power control unit 109e could reliably close memory write processes in less than 2 milliseconds, in order for SRU 109 to shut down without inducing errors in memory 109a or RAM 213 operating as NVRAM. However, under preferred operating conditions, SRU 109 receives both (i) a shutdown command from device 101 (equivalent to a “remove” or “dismount” command from device 101 for SRU 109) and (ii) at least 10s of milliseconds of power on pin 4 201g before device 101 terminates power input and supplied voltage to pin 4 201g. Power capacitor 208 could be utilized if power is suddenly removed from SRU 109 without a shutdown command from device 101. Note that power capacitor 208 could be used with power control unit 109e to smooth voltage fluctuations received through pins 109x, such that a more stable voltage can be supplied to components connected via internal bus 212.

In exemplary embodiments, power control unit 109e delays the completion of the startup of SRU 109 until after power capacitor 208 is sufficiently charged, such as being charged to a level of > 75% of capacity. In these exemplary embodiments, the delay of startup of SRU 109 from the application of power to SRU 109 from device 101 can normally be delayed to less than a few tens of milliseconds. The benefit of delaying startup of SRU 109 until power capacitor 208 is charged is that fluctuations in power from device 101 can be smoothed out, such as a transient power applied of a few milliseconds would not start SRU 109 in a state where SRU

109 cannot be cleanly shut down through the operation of power capacitor 208. Note that “power capacitor” herein contemplates a capacitor of sufficient size to power SRU 109 for at least a few milliseconds, which is different than a power capacitor for larger electrical circuits such as power supplies, which could use capacitors on the order of farads. Further, upon a power source with device 101 being removed, power control unit 109e could shut down radio 109b before shutting down processor 109c, while using stored electrical energy from power capacitor 208 in order to reduce current drain and allocate power resources to processor 109c cleanly shutting down processes when device 101 removes power from SRU 109.

Many of the logical steps for operation of SRU 109 and RU 109b can be performed in software and hardware by various combinations of processor 109d, processor 109c, firmware 109u or boot firmware 217 and other components in Figure 2a. When SRU 109 or RU 109b is described herein as performing various actions such reading a file, writing a file, transmitting or sending data, receiving data, encrypting a ciphertext, conducting a key exchange, etc., specifying herein that SRU 109 or RU 109b performs an action can refer to software, hardware, and/or firmware operating within SRU 109 illustrated in Figure 2a performing the action.

Figure 2b

Figure 2b is a graphical illustration of an exemplary system, where a device and a radio unit communicate through a shared nonvolatile memory in a storage radio unit, in accordance with exemplary embodiments. Device 101 and radio unit (RU) 109b can preferably communicate through an interconnected electrical interface and mutually accessible nonvolatile memory 109h in storage radio unit 109, in order to transfer data between the two elements. Exemplary embodiments for a system supporting communication through a shared nonvolatile memory 109h is illustrated in Figure 2b. Device 101 and RU 109b can both mutually access a shared memory 109h, which could comprise a selected portion of memory 109f, where memory 109f can comprise a selected portion of overall nonvolatile memory 109a as depicted and described in connection with Figures 1c and 2a above. A system 200b supporting a shared nonvolatile memory 109h can comprise a device 101, a shared memory 109h, an SRU external bus 202, and SRU internal bus 212, RU polling 232, device polling 233 and RU 109b. For some embodiments, mutually accessible nonvolatile memory 109h or shared nonvolatile memory 109h could operate in device 101 outside of a SRU 109, such as depicted in Figure 7 below.

A shared nonvolatile memory 109h can operate as a shared nonvolatile memory interface 230. A shared nonvolatile memory interface 230 can be referred to herein as a shared memory interface 230, where the shared memory interface 230 can operate in a nonvolatile memory

(such that data stored in the nonvolatile memory is recorded when power is removed from the memory cells recording the data). As contemplated herein, a shared memory interface 230 can be referred to as a “shared filesystem interface” 230’, for embodiments where shared nonvolatile memory 109h is formatted or structured with a filesystem for both device 101 and RU 109. In other words, a shared filesystem interface 230’ can be a subset or specific configuration for a shared memory interface 230, where a shared filesystem interface 230’ records a filesystem in a shared nonvolatile memory interface 230.

Note that the use of a shared filesystem interface 230’ as the structure for shared nonvolatile memory 109h is not required, and other structures are possible as well, such as simply addressing specific memory cells by both device 101 and RU 109b for the data depicted in Figure 2b. Some embodiments may use a shared filesystem interface 230’ within shared nonvolatile memory 109h in order to “reuse” widely adopted technology and existing, compatible technology of filesystem in device 101 and SRU 109 for addressing, reading, and writing to storage nonvolatile memory. Shared filesystem interface 230’ can also include a configuration process 231 and a device write and radio unit read memory 191 and device read and radio unit write memory 192. As contemplated herein, a “shared filesystem interface” 230’ may also be described or referred to as “shared filesystem” 230’.

The use of a shared filesystem 230’ for shared memory 109h can be desirable to establish communication between device 101 and RU 109b when (i) RU 109b resides on a storage radio unit 109 and (ii) SRU 109 operates with (i) established standards for removable media or (ii) external storage from CPU 101b, such as (a) using generic removable media drivers for device driver 101g and/or (b) using established filesystem and device drivers in device 101 for memory 109h and filesystem 230. The generic or pre-existing device driver 101g installed on a wide variety of devices 101 (or pre-included in various OS 101h for many different devices 101) may not include specific interfaces for communication with RU 109b operating inside a SRU 109, but communication between device 101 and RU 109b can be desirable in exemplary embodiments. For example, electrical pins 109x and device driver 101g could support a one-bit SD bus for removable media such as a micro SD card and resulting file or memory operations within the micro SD card (via SD card driver 101k), and SD card driver 101k included in device 101 may not require specific support to send data directly to RU 109. Data reporting libraries 101x could include the logic of using a standard device driver 101g such as an SD card driver 101k to write file or data to nonvolatile memory 109a, which SRU 109 could structure as a shared memory 109h, such that RU 109b could read the data written by data reporting libraries 101x.

The manufacturer of an SRU 109 may incur high commercial expenses in order to create device drivers 101g for the very wide variety of devices 101 and device OS 101h, as described in the needs of the art detailed above. However, the manufacturer of an SRU 109 could more quickly and for lower cost establish compatibility with a wide variety of devices 101 using an SD card driver 101k interface for SRU 109. Note that other common drivers besides SD card driver 101k could be utilized, such as a USB interface, PCIe interface, SPI interface, SATA interface, serial interface, etc.

Other possibilities exist as well, where a wide variety of manufactured devices 101 can (A) support standards-based storage media such as SD cards, but (B) lack pre-installed drivers 101g with protocols for communicating directly with RU 109b through a shared nonvolatile memory 109h. Storage radio unit 109 and RU 109b could subsequently utilize system 200b in order to communicate with device 101, where support for a device 101 to read and write to non-volatile memory 109a, memory 109f, and memory 109h could be widely supported by existing devices 101 and device OS 101h. Further, device 101 could use system 200b in order to communicate with radio unit 109b using a system 200b and a shared filesystem interface 230'.

A shared memory 109h with a shared filesystem interface 230' can be compatible with a generic or widely distributed media device driver 101g (such as SD card driver 101k). Use of a standard media device driver 101g can allow device 101 to (i) write data to memory 191 or files in storage radio unit 109, (ii) read data o from memory 192 or files in storage radio unit 109, and (iii) quickly and reliably mount or attach the shared memory 109h within an SRU 109. Using system 200b, RU 109b and device 101 can subsequently utilize this standard capability to communicate and share data, while SRU 109 also conducts or performs the function of nonvolatile memory storage for device 101.

In other words, for device 101 to send data to RU 109b in an exemplary embodiment, device driver 101g can write the data to memory 191 in shared memory 109h, where RU 109b can subsequently read data in the memory 191 in shared memory 109h using read libraries 217a within memory control libraries 217. Other, similar embodiments are possible as well, such as RU 109b using memory read and writes libraries in SRU firmware 109u (in which case read and write libraries in EEPROM 217 would be for boot functions). Or, SRU 109 using memory processor 109d and firmware 109u could conduct a read of memory 191 and send the data to RU 109b via internal bus 212.

RU 109b with processor 109c can conduct periodic polling 232 of files in memory 191 within shared memory 109h to determine that device 101 has written data for RU 109b to subsequently read (e.g. polling can detect if the timestamp for a file has been changed or

updated since a previous polling, or if a new file is available for reading by RU 109b). Further, in order for device 101 to read data from RU 109b, RU 109b could write the data to a file system in memory 192 in shared memory 109h using write libraries 217b within memory control libraries 217. Device 101 could subsequently read the data from memory 192 in shared memory
5 109h using device driver 101g. Device 101 can use periodic device polling 233 of memory 192 in shared memory 109h to determine that RU 109b has written data for device 101 to subsequently read.

As contemplated herein, RU polling 232 or device polling 233 could comprise a logical loop operating with processor 109c or processor 101b, respectively, where the loop determines
10 that a file or value recorded in memory has been updated or changed by the other node (where device 101 and RU 109b can comprise the two nodes). The loop timing for RU polling 232 or device polling 233 could operate at specified frequencies, such as an exemplary every 10 or 50 milliseconds, although other possibilities exist as well without departing from the scope of the present disclosure. In addition, other mechanisms exist for SRU 109 to use instead of a polling
15 232, such as the use of a trigger library 245 depicted and described in Figure 2e below. With a trigger library or function, data from device 101 passes through memory/interface processor 109d before being written to memory 109h, and processor 109d could include a library with a “trigger” such that processor 109d notifies processor 109c and/or RU 109b if a file is being written from device 101 to memory 191 for RU 109b to read.

In an exemplary embodiment, the memory/interface processor 109d could be coupled to
20 the power control unit 109e. The normal or default state for radio unit 109b could be an off or a sleep state and radio unit 109b could be in the sleep state. Data from device 101 to be stored in memory 109h (such as application data 112a) could trigger memory/interface processor 109d to signal power control unit 109e to wake radio unit 109b. Radio unit 109b could transition from
25 the sleep state to a wake or operational state and subsequently read the application data 112a and transmit to the wireless network 102. Or, a processor 109c could operate the trigger instead of a memory/interface processor, such that processor 109c wakes radio unit 109b upon receipt of application data 112a.

Before recording shared data in shared memory 109h, (i) SRU 109 or (ii) a configuration
30 unit 110d for device service provider 110 or (iii) a configuration unit for SRU-SP 106 could perform a shared memory filesystem formatting and configuration process 231. The process 231 could comprise the standard steps for partitioning, formatting nonvolatile storage media such as an SD card, where a file system is applied to the available memory. Process 231 could include formatting memory 109h to standards supported by device 101 such as exemplary file

systems of FAT16, FAT 32, exFAT, NTFS, ext3, ext4, exFAT, JFFS2, or UDF. Other file systems for memory 109f and memory 109h are possible as well without departing from the scope of the present disclosure. Process 231 could also format all of non-volatile memory 109a (which can include shared memory 109h). Additional details regarding a process 231 are provided in Figure 2f below. At the conclusion of a configuration process 231 on shared memory 109h, device 101 and RU 109b can mutually access a shared filesystem 230', where the shared filesystem 230' resides within storage radio unit 109 that also operates radio unit 109b.

In other words, device 101 or device service provider 110 can normally be responsible for selecting the format of a file system of memory 109g for general files 109g used by device 101. During a configuration process or step 231, the first portion of configuration 231 can comprise selecting, formatting, and writing a regular file system format for standard storage nonvolatile memory 109g or storage media, including removable media. In exemplary embodiments with the use of inserted media in device 101 such as, but not limited to, a SD card or micro SD card, the regular file system format can support existing SD cards that may not include a RU 109b in SRU 109. A second portion of configuration process 231 can comprise SRU 109 subsequently detecting and utilizing the selected file system from the first portion of a configuration process 231 over memory 109g for device 101.

In this manner RU 109b within a storage radio unit 109 can support the use of existing device drivers 101g in devices 101 (where device drivers 101g in device 101 are known to be compatible with the file system in memory 109g since memory 109g is structured to be read by device 101 using device drivers 101g). Communication between device 101 and RU 109b using a system 200b can utilize a device program 101i depicted and described in connection with Figure 1b for a device 101 (such as application data reporting libraries 101x) (i) to write data to memory 191 in shared memory 109h, (ii) read data from memory 192 in shared memory 109h, and (iii) conduct a periodic device polling 233. Additional details regarding a configuration process 231 are depicted and described in connection with Figure 2f below.

After a configuration 231 step, either device 101 or SRU 109 can load shared memory 109h with an initial set of values or files supporting the operation of RU 109b in SRU 109. In exemplary embodiments, the initial set of values or files for (x) shared memory 109h and a shared filesystem 230' could be included in (y) SRU firmware 109u. Additional details for firmware or libraries to support a shared filesystem 230' in shared memory 109h are depicted and described in connection with Figure 2e below. Device 101 could write data into device write and SRU read memory 191 in the format of either (i) multiple different files, or (ii) a single file with multiple values recorded in the file.

The file or files in memory 191 could include the depicted values of radio settings 191a, http(s) request/URL 191b, “mobile originated” (MO) SMS 191c, upload file 191d, cryptographic parameters 191e, signaling data 191f (to RU 109b), and control data 191g (for RU 109b). RU 109b could write data into RU write and device read memory 192 in the format of either (i) multiple different files, or (ii) a single file with multiple values recorded in the file. The file or files in memory 192 could include the depicted values of radio configuration 192a, http(s) response/file 192b, “mobile terminated” (MT) SMS 192c, download file 192d, cryptographic settings 192e, signaling data 192f (to device 101), and control data 192g (for device 101). Other possibilities exist as well for the values or data recorded in shared memory 109h without departing from the scope of the present disclosure, and the exemplary data above is illustrative as opposed to limiting. “Http(s)” shows that the data could be for transmission as either hypertext transfer protocol (http) or secured hypertext transfer protocol (https) such as using TLS or DTLS.

As an example of the data flow supported by a shared filesystem interface 230' in a system 200b, after configuration process 231, device 101 could write a MO SMS to 191c. RU 109b could conduct polling 232 in filesystem interface 230' within shared memory 109h to detect that a new MO SMS from device 101 is written to 191c. Power control unit 109e could power up components from a powered down or deep sleep state to a fully powered or operational state with RF front end 203, RF transceiver 204, eUICC 113, and baseband unit 206, such that RU 109b could connect with wireless network 102. RU 109b can subsequently read MO SMS from 191c and transmit the MO SMS to wireless network 102. After transmission of MO SMS to wireless network 102, power control unit 109e could power down for force components within RU 109b from the previous sentence into a powered off or deep sleep state if no inbound data for device 101 is received within an exemplary time such as 10 seconds (although other time values are possible as well, such as 3 or 15 seconds). The time settings for power down after transmission of the MO SMS by RU 109b could be specified in radio settings 191a. Other possibilities exist as well for both (i) the values in shared memory 109h and (ii) the steps of transferring the data between device 101 and RU 109b (and transfer of data to wireless network 102 and/or SRU-SP 106) without departing from the scope of the present disclosure.

Figure 2c

Figure 2c is a graphical illustration of a storage radio unit utilizing two clock inputs, in accordance with exemplary embodiments. SRU 109 can include a SRU memory/interface processor 109d in order to communicate with both (i) device 101 using an external bus 202 and

(ii) internal components of SRU 109 using an internal bus 212. As contemplated herein, and “external bus” 202 can comprise an external-facing bus 202 such as the electrical pins 109x, while external bus 202 may otherwise reside within SRU 109. Figure 2c depicts a solution to the needs in the art described above in the section “Description of Related Art”, where an SRU 109 may operate more efficiently, faster, and more reliably from using an internal clock source that is different than an external clock from device 101, where an external clock is input via a pin such as pin 5 for the micro SD card physical interface. The sources of timing for a memory/interface processor 109d depicted in Figure 2c can be utilized with other interfaces as well, such as, but not limited to, an SPI, PCIe, USB, serial interface, or enhanced SPI (eSPI).

In exemplary embodiments, a physical interface 202 between SRU 109 and device 101 can utilize a clock input 201e from the “master” or “host” (e.g. device 101). The flow of signals or electrical pulses between SRU 109 and device 101 can be timed based on clocking signals such as CMOS square waves on clock input 201e. These clock signals could be in exemplary ranges of 10 – 50 MHz for an SD bus interface operating on bus 202, where the speed is determined by device 101 and derived from device clock 101m. The frequency of clock input 201e can comprise clock frequency 228, which can be input into processor 109d in order to operate interface driver 109y. In other words, interface driver 109y can be driven with the clock input 201e at a clock frequency of 228.

Although clock input 201e for SRU 109 can be required in order to transfer data between SRU 109 and device 101 over bus 202, the frequency and stability of clock input 201e can be far from desirable for an optimal operation of SRU 109. For example, although clock input 201e could be suitable for communication via bus 202 and pins 109x, clock input 201e could have relatively high jitter, noise, phase variation, etc. Using a relatively “noisy” clock input 201e in SRU 109 to drive internal bus 212 and internal components in SRU 109 (such as RAM 213, memory core interface 109b, etc.) could induce a higher level of errors for components in SRU 109 and connected via internal bus 212.

A typical clock input 201e from a low-cost device could have typical phase noise at the input frequency of -40 to -45 dBc/Hz. Note that significant “spikes” or “timing signal drops” much worse are possible, depending on the operation and power states of components in device 101, and these spikes of poor clock timing signals on clock input 201e could induce errors SRU 109. In contrast, internal RF clock 205 for SRU 109 could be a VTCXO or TCXO with a typical phase noise of -55 dBc/Hz at the output frequency. Note that SRU 109 controls RF clock 205 and thus can operate with a clock source with many fewer “spikes” or “drops”. Although the timing output of RF clock 205 may be set for RF transceiver 204 (such as an

exemplary 38.4 MHz with 4G LTE NB-IoT or CATM1), PLL 211 can scale clock 205 with low noise to a level such as an exemplary 192 MHz, where the output of PLL 211 can drive processor 109d as depicted in Figure 2c. The frequency of the output of PLL 211 as an input to processor 109b can comprise clock frequency 229, which can be an independent numerical value from clock frequency 228.

In exemplary embodiments and as depicted in Figure 2c, SRU memory/interface processor 109d can utilize clocking signal output from PLL 211 at a clock frequency 229 in order to drive (i) internal clock signals within processor 109d (excluding interface driver 109y) and (ii) internal bus 212 with internal bus driver 212a, and (iii) SRU 109 components attached to bus 212. In this manner, SRU 109 can control and/or reduce phase noise, “spikes” and “drops” in clocking signal timing from clock input 201e. Thus in exemplary embodiments, SRU processor 109d can utilize two clock inputs with two different frequencies, where the first clock input drives external facing bus 202 and the second clock input drives processor 109d and components on internal bus 212. In this manner, errors for communications and signals within processor 109d and electrical components in SRU 109 such as RAM 213 and nonvolatile memory 109a can be reduced.

Figure 2d

Figure 2d is a graphical illustration for using an interface between a device and a storage radio unit, in accordance with exemplary embodiments. Other interfaces besides interfaces for SD cards and micro SD cards can be used between device 101 and storage radio unit 109. System 200c in Figure 2d depicts the use of a serial peripheral interface (SPI), and other possibilities exist as well for the physical interface between device 101 and SRU 109 without departing from the scope of the present disclosure. In other words, the use of a filesystem interface 230' in shared memory 109h is not required for some exemplary embodiments, and the benefits of a system 100 in Figure 1, system 200a in Figure 2a, system 200b in Figure 2b, and system 400 in Figure 4, can be obtained without the use of a filesystem interface for a shared memory interface 230.

Device drivers 101g can include an SRU driver 234, where SRU driver controls the operation of SRU 109. SRU driver 234 could comprise a driver similar to a modem or module driver for a typical radio module for devices, with the addition of also controlling files in a nonvolatile memory of 109a of SRU 109. In other words, SRU 109 can include a radio and nonvolatile memory for device 101, and SRU driver 234 can comprise a driver using conventional technology for SRU 109 (e.g. a specific driver for the device OS 101h). When

using an SPI interface in system 200d, device 101 can also include an SPI master driver 235. SPI master driver 235 can include both an SPI memory library 235a and an SPI radio AT command library 235b.

SPI memory library 235a can support a first, memory interface 239a to memory core interface 109m operating in SRU 109. The memory interface 239a can pass through an SPI physical interface 236 for device 101, external pins 109x for storage radio unit 109, and SPI slave driver 237 in SRU 109 in order to communicate data with memory core interface 109m. Processor 109d can also be in the flow and control data sent and received over memory interface 239a between device 101 and memory core interface 109m. Device 101 can write and read data in nonvolatile memory 109a using memory interface 239a, where memory interface 239a uses an SPI interface. Other interfaces besides an SPI interface could be used as well, such as, but not limited to, a PCIe interface, serial interface, etc.

SPI radio AT command library 235b can support a second, AT interface 239b to radio unit 109b. The AT interface 239b can pass through an SPI physical interface 236 for device 101, external pins 109x for storage radio unit 109, and SPI slave driver 237 in SRU 109 in order to communicate data with AT command core 238. Processor 109d can also be in the flow and control data sent and received over memory interface 239b between device 101 and AT command core 238. AT command core 238 can drive and control radio unit 109b with conventional technology for an AT command set using AT command stack 109s, which is typical and standard for radio units, modules, and modems as of 2018. Device 101 can write AT commands to radio 109b via AT interface 239b and receive responses from radio 109b via AT interface 239b. In other words, AT interface 239b can operate as a traditional SPI interface for SRU 109 in some exemplary embodiments. Although an SPI interface is depicted in Figure 2d, other physical interfaces could be supported using the components and logic shown, such as an I2C, PCIe, PCI, UART, or serial interfaces, and other possibilities exist as well. An AT interface 239b and/or memory interface 239a could use a point-to-point protocol at a higher level than the physical layer in order to support communication between a device 101 and storage radio unit 109b.

Figure 2e

Figure 2e is a graphical illustration of storage radio unit firmware 109u, in accordance with exemplary embodiments. SRU firmware 109u can be recorded in nonvolatile memory 109r when SRU 109 is powered off, and an initial set of SRU firmware 109u could be loaded into SRU 109 by a manufacturer of SRU 109, distributor of SRU 109, SRU-SP 106, or possibly

device service provider 110. As described above in connection with Figures 1c and Figure 2a, SRU firmware 109u can be loaded by SRU 109 using boot firmware 217 and a boot configuration 218 recorded in EEPROM 216. SRU firmware 109u can include shared memory control libraries 240 for shared memory 109h, radio unit firmware 222 for radio unit 109b, an eUICC driver 249 for eUICC 113, an SD interface driver 109y for bus 202 and electrical pins 109x, SRU operating configuration 249b for SRU 109, and an operating system for SRU 109 comprising SRU OS 220. Although an SD interface driver 109y is depicted in Figure 2e, the interface driver could support other bus types communicating through electrical pins 109x.

Radio unit firmware 222 for radio unit 109b can include instructions and executable machine code and/or software libraries for processor 109c with radio unit 109b. Radio unit firmware 222 can include libraries for the operation of radio unit 109b, including an operating system for radio unit 109b. Radio unit firmware 222 can include firmware components to control and/or operate baseband unit 206, RF transceiver unit 204, RF front end 203, and interface with power control unit 109e and internal bus 212. Radio unit firmware can also include the logic and software/firmware for radio unit 109b to attach to wireless network 102 and transmit/receive data with wireless network 102 and also SRU-SP 106.

eUICC driver 249a can comprise instructions for radio unit 109b and processor 109c to communicate with eUICC 113. eUICC driver 249a can support an ES8 interface and data between eUICC 113 and an SM-DP (through radio unit 109b and wireless network 102), an ES5 interface and data between eUICC 113 and an SM-SR 107 (through radio unit 109b and wireless network 102), an ES10 interface and data between eUICC 113 and a local profile assistant operating in device 101 or SRU 109. In some exemplary embodiments, the communication of an ES10 interface for eUICC 113 can be with a server external to device 101, such as within SRU-SP 106. ES8 and ES5 interfaces and data are defined in GSM technical specification SGP.02. ES10 interface and data are specified in GSM technical specification SGP.22. eUICC driver 249a can also support communication between processor 109c in radio unit 109b and eUICC 113 such that eUICC 113 can operate with profiles 113c and 113d with network credentials and network parameters similar to a traditional SIM card in order for radio unit 109b to authenticate and connect with wireless network 102.

Interface driver 109y can comprise a driver for SRU 109 to communicate with device 101 via electrical pins 109x, and could comprise an SD bus driver for SRU 109 for SD card driver 101k in device 101. Other bus types and interfaces are possible as well for an interface driver 109y and driver 101k without departing from the scope of the present disclosure, and in exemplary embodiments driver 109y and driver 101k are of the same type and are compatible,

as well as the electrical signals through electrical pins 109x. Drivers 109y and 101k are also described above, such as with Figure 1b for device 101 with driver 101k and Figure 1c and Figure 2a for driver 109y.

SRU operating configuration 249b can comprise data, a file, or instructions for the operation of SRU 109. SRU operating configuration 249b can specify (i) the bus types for bus 202 and bus 212, (ii) values and settings for SRU 109 and radio unit 109b including timers, frequencies to utilize, memory allocations and settings for nonvolatile memory 109a, (iii) initial or power on settings for SRU 109, (iv) a list of supported capabilities for SRU 109, and (v) filesystem types, parameters, and partition information for data recorded in memory 109a, (vi) PLL 211 settings and clock settings, (vii) the settings or selection of a switch 203a to select an antenna for use by SRU 109. SRU operating configuration 249b can comprise a file such as an XML file, text file, and other possibilities exist as well without departing from the scope of the present disclosure.

In exemplary embodiments where SRU 109 includes a NVRAM for RAM 213 (e.g. in Figure 5a), after boot or an initial power up, SRU 109 records SRU operating configuration 249b in RAM 213 (shown in Figure 2a). RAM 213 can comprise an NVRAM, such that upon a subsequent power on, SRU 109 can read operating configuration 249b quickly from RAM 213 as NVRAM without first having to read and extract SRU operating configuration 249b from firmware 109u or flash nonvolatile memory 109r. SRU OS 220 can comprise an operating system suitable for SRU 109. Example operating systems 220 for SRU 109 include eCos, uC/OS, LiteOs, Contiki, OpenWRT, Raspbian, and other possibilities exist as well without departing from the scope of the present disclosure.

Shared memory control libraries 240 for shared memory 109h in SRU firmware 109u can include filesystem detection libraries 241, filesystem drivers 242, shared memory radio unit read library 243, device write directory filesystem structure 191x', device write directory trigger library 245, shared memory radio unit write library 246, and device write directory filesystem structure 192x'. Filesystem detection libraries 241 can read, detect, and determine supported filesystems in device general memory 109g such as filesystem 229 (shown in Figure 2a), where device general memory 109g can be configured, formatted, and/or partitioned by device 101, device service provider 110, or a user of device 101. Device general memory 109g can be configured by a configuration unit 110d or by device 101 directly. In exemplary embodiments, memory 109g can be configured in SRU 109 with a default file system such as a FAT 32 or an exFAT filesystem, although device 101 or a configuration unit 110d could select and write a different filesystem for memory 109g after distribution of SRU 109. Filesystem detection

libraries 241 can detect and determine a filesystem for memory 109g comprising filesystem 229 according to all available and supported filesystems in filesystem drivers 242. The use of a filesystem detection library 241 is also depicted and described in connection with step 255 in Figure 2f below.

5 Filesystem drivers 242 can comprise kernel modules for OS 220 in order for SRU 109 to access, read, and write files and data to memory 109f and memory 109r. Since device 101 can specify, format, and control a filesystem 229 in memory 109g within memory 109f, SRU 109 can preferably support a wide variety of filesystem drivers 242 in order for SRU 109 to read and write data and files for memory 109f, where the files and data can be subsequently be
10 compatibly accessed by device 101.

In other words, with conventional technology, a storage unit such an SD card with flash memory storage and a processor 109d and memory core interface 109m may operate with physical level or layer data and not operate with internal filesystem detection libraries 241 and filesystem drivers 242 in order for the SRU to write data (a) internally generated by the SRU
15 109 (or received by an NB-IoT network 104 or wireless network 102) and (b) for the device 101 to subsequently read. The source of data input for recording data in memory 109a is device 101 with conventional technology, while for exemplary embodiments the source of data input for recording data in memory 109a is internally generated by SRU 109.

Filesystem drivers 242 can support any standard filesystem used by a device 101 with
20 memory 109g and memory 109f, such as, but not limited to the filesystem types of FAT16, FAT32, exFAT, NTFS, Ext2, Ext3, Ext4, UDF, ISO-9660, HFX+, HFSX, Apple, UFS, XFS, JFS, YAFFS, JFFS2, UBIFS, and other possibilities exist as well, including subsequent, future, or related filesystems for these exemplary filesystems of a filesystem 229. In exemplary embodiments, SRU 109 can support a plurality of filesystem drivers 242 such as those depicted
25 in Figure 2f and select an individual filesystem driver 242 for OS 220 in SRU 109 to read and write files to memory 109h based on a detected filesystem for device 101 in memory 109g. In exemplary embodiments, not all depicted filesystem drivers 242 within SRU 109 are required, and a subset of the filesystems could be selected for SRU 109.

Read libraries 243 can provide machine executable code for processor 109d to read data
30 from device 101 using shared memory 109h. Read libraries 243 can be selected from filesystem drivers 242 to be compatible with filesystem 229 used by device 101 for memory 109g. The data read from shared memory 109h by read libraries 243 could be used with RU 109b or input into RU 109b. In this manner by using read libraries 243, memory 109h can provide the physical and logical interface for RU 109b to receive data from device 101. Read libraries 243

could specify memory addresses or file locations in a shared file system 230 for non-volatile memory 109h where device 101 can write data in order to be read by SRU 109 and RU 109b. In an exemplary embodiment, (i) device 101 could write a file with a name "http request" 191a (such a URL with parameters for server B 110a) to a specified location or file name in shared memory 109h, such as the depicted memory for device write and radio unit read memory 191 in Figure 2b above, and then (ii) read libraries 243 could instruct processor 109c or processor 109d to read the data in 191 and subsequently use data 191 (which could comprise the exemplary http request 191a) for transmission through radio unit 109b to SRU-SP 106 and/or server B 110a.

In exemplary embodiments, either (a) processor 109d could operate read libraries 242 and pass the data from memory 191 to processor 109c in radio unit 109b, or (b) processor 109c could operate read libraries 242 and read from memory 191 directly in order to pass the data 112a to wireless network 102. Other possibilities exist as well for an SRU 109 to read data input from device 101 into a shared filesystem 230' in storage radio unit 109 without departing from the scope of the present disclosure. Device write directory filesystem structure 191x' in device 101 (shown in Figure 1b) can specify the names of files and/or settings within a directory of shared filesystem interface 230' for device 101 to utilize in order for device 101 to send data to radio unit 109b. Device write directory filesystem structure 191x' can comprise the structure of a device write directory 191x with pre-agreed file names for data input from device 101 into SRU 109 for RU 109b. For example, the recoding the filesystem structure 191x' by device 101 is depicted and described for device 101 in Figure 1b. An exemplary device write directory 191x using a pre-agreed filesystem structure 191x' is depicted and described in connection with Figure 3a below.

Write libraries 246 can provide machine executable code for processor 109d to write data output from radio unit 109b to memory 109h and shared filesystem interface 230' in order for device 101 subsequently read for data input into device 101. Write libraries 246 can be selected from filesystem drivers 242 to be compatible with filesystem 229 used by device 101 for memory 109g. Note that filesystem 229 by device 101 can also be used as the filesystem for memory 101h. In this manner, write libraries 246 could provide the logical software or firmware interface for RU 109b to send data received from wireless network 102 to device 101, and other possibilities exist as well for the transfer of data from RU 109b to device 101 through a shared filesystem interface 230' in shared memory 109h without departing from the scope of the present disclosure.

Write libraries 246 could specify memory addresses or file locations in a file system 230 for non-volatile memory 109h where RU 109b using processor 109c can write data in order to

be subsequently read by device 101. In an exemplary embodiment, (i) RU 109b could write a file with a name "http response.htm" 192a (such as data received from server B 110a and/or SRU-SP 106 via wireless network 102) to a specified file location in memory 109h, such as the depicted memory 192 for device read and radio unit write operations, and then (ii) device 101
5 could subsequently read the data in 192 via bus 202 in order to obtain the server B response 112b data depicted in Figure 1a.

In exemplary embodiments, either (a) processor 109d could operate write libraries 246 and pass the data from processor 109c in radio unit 109b to memory 192, or (b) processor 109c could operate write libraries 246 and write to memory 192 in shared memory 109h directly in
10 order to pass the data 112b from wireless network 102 to device 101. Other possibilities exist as well for an SRU 109 to write data 112b output for device 101 into a shared filesystem 230' in storage radio unit 109 without departing from the scope of the present disclosure. Device read directory filesystem structure 192x' can specify the names of files within a directory of shared filesystem interface 230' for device 101 to utilize in order for device 101 to read data from radio
15 unit 109b. Device read directory filesystem structure 192x' can comprise settings for a directory with pre-agreed file names for data input from RU 109b (and/or SRU 109) into shared memory 109h or shared filesystem 230' for device 101 to subsequent read as data 112b input into device 101.

Note that device 101 can also store and operate with a device read filesystem structure
20 192x' in device program 101i including data reporting libraries 101x. An exemplary device write directory filesystem structure 192x is depicted and described in connection with Figure 3b below. In exemplary embodiments the separate read and write functions in SRU firmware 109u in Figure 2e and separate read and write memory in shared memory 109h could be combined into a single read/write function in SRU firmware 109u and a single read/write memory or set of
25 files in shared memory 109h.

In an exemplary embodiment described above for Figure 2e, SRU 109, using read libraries 243 and write libraries 246, can (i) read data from device 101 using shared memory 109h and (ii) write data to memory 109h, where device 101 can subsequently read data from memory 109h. In another embodiment, write libraries 246 could provide instructions for SRU
30 to write data 112b directly to RAM 101e or processor 101b, but in this case a closely coupled software and firmware interface between SRU 109 and device 101 through electrical pins 109a and/or bus 202 would be required. For example an SPI bus mode, a one-bit SD bus mode, or a four-bit SD bus mode for the interface through electrical pins 109a would not normally be able to support SRU 109 writing data directly to RAM 101e or processor 101b without instruction

from device 101 (e.g. SRU 109 with conventional technology is a “slave” or “client” and device 101 is a “master” or “host”), so alternative bus technology connecting storage radio unit 109 and device 101 could be utilized for SRU 109 to write data directly to RAM 101e or processor 101b (without instruction from device 101).

5 Trigger library 245 can comprise executable machine code for SRU 109 and processor 109d to automatically detect when device 101 writes data to shared memory 109h and/or shared filesystem 230' with device write memory 191. Processor 109d using a trigger library 245 could provide an alternative to RU 109b polling 232 depicted in Figure 2b above, and in this manner RU polling 232 could be bypassed and processor 109d could signal RU 109b when data is
10 available in memory 191 for RU 109b to read, process, and/or transmit to wireless network 102. In exemplary embodiments, trigger library 245 is coupled with power control unit 109e (from Figure 1c and Figure 2a above), such that power control unit 109e (a) instructs RU 109b to enter a sleep mode (without requiring polling 232) when no new or “untransmitted” data 112a is available from device 101 in memory 109h and shared filesystem 230', and (b) instructs RU
15 109b to wake or enter an active mode when new data 112a has been written by device 101 to memory 109h as detected and processed by a trigger library 245. After a confirmation or response data 112b is received by SRU 109 and written to shared memory 192, power control unit 109e could instruct radio unit 109b to enter a sleep or deep sleep mode or state, such as consuming less than an exemplary 5 microamps.

20 Note that the use of a trigger library 245 and a power control unit 109e in conjunction can help minimize the time radio unit 109b remains in an active state, where data 112b can be recorded in nonvolatile memory 192 after receipt by RU 109b from wireless network 102, and RU 109b can consequently enter a sleep mode without having to wait for device 101 or device CPU 101b to read the data 112b. Device 101 can read the server/network response data 112b
25 from memory 192 in memory 109h at a time that is convenient and/or optimized energy utilization for device 101 and CPU 101b. In an exemplary embodiment, device 101 could enter a sleep state while RU 109b is sending data 112a and receiving data 112b (as depicted and described in connection with Figure 4a below), and SRU 109 would prefer to not wait until device 101 wakes in order to pass server B response data 112b to device 101.

30 In other words, through the use of a nonvolatile memory 192 to store or record (a) server/network response data 112b in response to (a) device 101 transmitted data 112a, the amount of time that CPU 101b is active and RU 109b is active can be reduced compared to conventional technology, since the communication between CPU 101b and RU 109b can take place through a nonvolatile shared memory interface. Likewise, for device transmit data 112a

(e.g. application data 112a), CPU 109b in device 101 could write the data in memory 191 for memory 109h while RU 109b sleeps. Then, CPU 109b can essentially immediately enter a sleep state and remain in a sleep state until (i) RU 109b writes server B response data 112b in memory 192 and SRU 109 can (ii.a) send a signal via pins 109x for CPU 109b to wake and read data 112b from memory 192 and (ii.b) sleep RU 109b essentially immediately after data 112b is written to memory 192 in shared memory 109hs.

Figure 2f

Figure 2f is a flow chart illustrating exemplary steps for configuring a storage radio unit to communicate with a device using a shared nonvolatile memory interface, in accordance with exemplary embodiments. In order for storage radio unit (SRU) 109 to communicate with device 101 through a shared memory 109h and a shared nonvolatile memory interface 230 (which can comprise a shared filesystem interface) depicted described above, including in Figure 1c and Figure 2a, an SRU 109 can conduct a configuration process with steps depicted in Figure 2f.

The processes and operations, described below with respect to all of the logic flow diagrams may include the manipulation of signals by a processor and the maintenance of these signals within data structures resident in one or more memory storage devices. For the purposes of this discussion, a process can be generally conceived to be a sequence of computer-executed steps leading to a desired result.

These steps usually require physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared, or otherwise manipulated. It is convention for those skilled in the art to refer to representations of these signals as bits, bytes, words, information, elements, symbols, characters, numbers, points, data, entries, objects, images, files, or the like. It should be kept in mind, however, that these and similar terms are associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as listing, creating, adding, calculating, comparing, moving, receiving, determining, configuring, identifying, populating, loading, performing, executing, storing etc. that are often associated with manual operations performed by a human operator. The operations described herein can be machine operations performed in conjunction with various input provided by a human operator or user that interacts with the computer.

In addition, it should be understood that the programs, processes, methods, etc. described herein are not related or limited to any particular computer or apparatus. Rather, various types of general purpose machines may be used with the following process in accordance with the teachings described herein.

5 The present disclosure may comprise a computer program or hardware or a combination thereof which embodies the functions described herein and illustrated in the appended flow charts. However, it should be apparent that there could be many different ways of implementing the disclosure in computer programming or hardware design, and the disclosure should not be construed as limited to any one set of computer program instructions.

10 Further, a skilled programmer would be able to write such a computer program or identify the appropriate hardware circuits to implement the disclosed disclosure without difficulty based on the flow charts and associated description in the application text, for example. Therefore, disclosure of a particular set of program code instructions or detailed hardware devices is not considered necessary for an adequate understanding of how to make and
15 use the disclosure. The inventive functionality of the claimed computer implemented processes will be explained in more detail in the following description in conjunction with the remaining Figures illustrating other process flows.

Further, certain steps in the processes or process flow described in all of the logic flow diagrams below must naturally precede others for the present disclosure to function as described.
20 However, the present disclosure is not limited to the order of the steps described if such order or sequence does not alter the functionality of the present disclosure. That is, it is recognized that some steps may be performed before, after, or in parallel other steps without departing from the scope and spirit of the present disclosure.

The processes, operations, and steps performed by the hardware and software described
25 in this document usually include the manipulation of signals by a CPU or remote server and the maintenance of these signals within data structures resident in one or more of the local or remote memory storage devices. Such data structures impose a physical organization upon the collection of data stored within a memory storage device and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art
30 of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

At step 251, SRU 109 can be manufactured w/ radio unit 109b, SRU ID 109i, nonvolatile memory 109a, SRU only accessible memory 109r, and firmware 217, 218, and 109u. Other electrical components for an SRU 109 depicted in Figure 2a could be included in a

manufacturing step 251 as well. SRU firmware 109u as depicted in Figure 2e above can include an operating system 220 and filesystem drivers 242 for SRU 109, as well as the other data depicted for SRU firmware 109u. An SRU-SP 106 could then receive the manufactured SRU 109 units from a step 251 and conduct a step 252 to write configuration data to SRU 109. Other entities such as a device service provider 110 or distributor of SRU 109 could conduct a step 252 as well. At step 252, SRU 109 records SRU private key SRU SK 109p, eUICC bootstrap profile 113c, and SRU-SP certificate with an SRU-SP public key 106b. For a step 252, SRU 109 could derive SRU SK 109p and SRU PK 109w, using the same steps and data to derive ephemeral PKI keys in a step 409 below in Figure 4 (but with different PKI keys output). SRU PK 109w could be written to memory 192 after a set of configuration step 231 format or structure the shared memory 192. SRU firmware 109u could be recorded in a step 252 instead of a step 251, and other possibilities for the sequence of steps depicted in Figure 2f are possible as well without departing from the scope of the present disclosure. A configuration unit similar to configuration unit 110d in Figure 1a could conduct step 252 with SRU 109.

At step 253, SRU 109 can be distributed and inserted into a device 101. SRU 109 could be inserted into device 101 during manufacturing of device 101, or after device 101 is distributed to users or organizations owning device 101, where the user or entity owning device 101 inserts SRU 109. For example, with embodiments where SRU 109 comprises a micro SD card form factor such as within a housing 227, SRU 109 could be inserted into device 101 in a step 253. SRU 109 could comprise an integrated circuit or wafer level chip scale package for inclusion in device 101, and SRU 109 could be soldered onto a circuit board or inserted into a larger wafer level package in step 253. SRU 109 could comprise a module or components with a larger, integrated circuit comprising a system on a chip 801 from Figure 8, where a step 253 can be including the components for an SRU 109 on the system on a chip 801. As depicted in Figure 2f, the collection of steps 251 through 253 can comprise a step 260.

At step 254 a filesystem 229 can be written to memory 109g in nonvolatile memory 109a of storage unit 109. The exemplary filesystem 229 could comprise any of the exemplary file systems depicted and described in connection with Figure 2e above. A configuration unit 110a for device service provider 110 could write the filesystem 229 to memory 109g. Or, device 101 could perform the steps of a filesystem 229 configuration on SRU 109, equivalent to a personal computer formatting a USB memory stick or SD card with a file system. Other possibilities for conducting a step 254 include a manufacturer of SRU 109 or SRU-SP 106 (i) creating a default filesystem for SRU 109, which could be an exFAT or FAT32 file system, and (ii) writing the default filesystem to memory 109g. For example, SRU-SP 106 could create filesystem 229

when data is written to SRU 109 in a step 252 above. General files for device 101 could also be written in a step 254, such as writing a device OS 101h within the filesystem 229. Other data, including device reporting libraries 101x and device drivers 101g can be recorded in the filesystem 229 in a step 254. Note that a step 254 can be conducted before a step 253, such that
5 memory 101g with filesystem 229 includes data for device 101 to operate before SRU 109 is inserted into device 101.

At step 255, SRU 109 can receive power from device 101 and boot with boot firmware 217 and boot configuration 218. SRU 109 can load firmware 109u in order to begin operating. SRU 109 can conduct a step 255 with filesystem detection libraries 241 and filesystem drivers
10 242 in order to determine and read from the filesystem 229 previously written to device general memory 109g in step 254. In other words, processor 109d can (i) read the locally stored files in memory 109a and memory 109g using filesystem detection libraries 241 in order to (ii) determine the type of filesystem 229. The determined or detected type of filesystem 229 from a step 255 could comprise any of the listed exemplary file systems from Figure 2d above. Note
15 that SRU 109 can conduct a step 255 without requiring any input from device 101 during a step 255, other than device 101 providing an electrical power source.

At step 256, SRU 109 can select a file system driver 242 based on the detected and determined file system 229 from a step 255. At step 257, SRU 109 can write SRU 109 device write directory 191x to shared memory 109h using the selected file system driver 242. A step
20 257 can also include SRU 109 conducting a filesystem formatting process, in order to write the filesystem 229 to memory 191 in shared memory 109h. Or, in a step 257, SRU 109 can write files for memory 191 in an existing filesystem detected for filesystem 229. In other words, SRU 109 can add a directory 191x to an existing filesystem 229, and the directory 191x can comprise device write memory 191. Thus, a directory 191x could comprise a directory added to an
25 existing filesystem 229, where the directory 191x could comprise a portion of the shared nonvolatile memory 109h.

SRU 109 can use (i) memory write libraries 217c from boot firmware and (ii) memory core interface 109m in order to change NAND and/or NOR flash memory gates, or memory blocks in order to write binary data to memory 109h, where the resulting binary data written can
30 comprise a device write directory 191x for the file system 229. Device write directory 191x at step 257 can also include base data such as the exemplary files and data depicted in Figure 3a below. The file permissions 301a for a device write directory (in Fig. 3a below) can be set at the same time as a step 257. In addition, user a 302a and user b 302b can be specified (shown in Figure 3a below). Further, a step 257 can comprise SRU 109 specifying that user a 302a for file

permissions 301a in device write directory 191x can be the same as user a 302a for filesystem 229 in general memory 109g for device 101. In this manner, device 101 can determine that OS 101h also has write access to directory 191x (since OS 101h or the equivalent user can be the user a 302a in a filesystem 229 for memory 109g).

5 At step 258, SRU 109 can write SRU write directory 192x to shared memory 109h using the selected file system driver 242. Or, in a step 258, SRU 109 can write files for memory 192 in an existing filesystem detected for filesystem 229. In other words, SRU 109 can add a directory 192x to an existing filesystem 229, and the directory 192x can comprise device write memory 192. SRU 109 can use (i) memory write libraries 217c from boot firmware and (ii)
10 memory core interface 109m in order to change NAND and/or NOR flash memory gates, or memory blocks in order to write binary data to memory 109h, where the resulting binary data written can comprise a RU write directory 192x for the file system 229. RU write directory 192x (or equivalently device read directory 192x) at step 258 can also include base data such as the exemplary files and data depicted in Figure 3b below. The file permissions 301a and 301b
15 for a RU write directory (in Fig. 3a below) can be set at the same time as a step 258. In addition, user a 302a and user b 302b can be specified.

Further, a step 258 can comprise SRU 109 specifying that user a 302a for file permissions 301a in RU write directory 192x can be the same as user a 302a for filesystem 229 in general memory 109g for device 101. In this manner, device 101 can determine that OS 101h
20 also has read access to directory 192x (since OS 101h can be the user a 302a in a filesystem 229 for memory 109g). As depicted in Figure 2f, the collection of steps 255 through 258 can comprise a configuration step 231, and a configuration step 231 is also depicted and described in connection with Figure 2b above. Configuration step 231 can also be referred to as a configuration process 231.

25 At step 259, device 101 driver 101k can record or store device write files in directory 191x, such as data from reporting libraries 101x which could also comprise application data 112a from Figure 1a above. As noted above in Figure 2e, CPU 101b can sleep after writing data to directory 191x, since the data is stored in a nonvolatile flash memory and RU 109b can read the data at a later time after CPU 101b enters a sleep state. Note that power to SRU 109 via
30 electrical pins 109x can continue after CPU 101b enters a sleep state. A power control unit 109e can detect new data in directory 191x, such as from a trigger in a trigger library 245, and power up and change RU 109b from a sleep state to an active state. After either (i) a RU polling 232 detection of new data in directory 191x or (ii) a trigger library 245 detection or recording of new data in directory 191x, RU 109b can subsequently read the data from directory 191x and process

the data. If the data written to directory 191x comprises application data 112a, then RU 109b can transmit the data to a wireless network 102.

Figure 3a

Figure 3a is a graphical illustration of a shared filesystem interface for a radio storage unit and a device, using a device write directory, in accordance with exemplary embodiments. SRU 109 can operate and record a device write and RU read memory 191, where memory 191 can reside within shared nonvolatile memory 109h (as depicted in Figure 2a). Shared nonvolatile memory 109h could be formatted with a shared filesystem 230' using the configuration steps 231, such that device 101 could write to shared filesystem 230' and RU 109b and SRU 109 could read from shared filesystem 230'. In addition, memory 191 can be referred to as "radio unit read and device write memory" 191 or simply memory 191. Shared filesystem memory 230' device write directory 191x is depicted as viewable and readable from device operating system 101h with a file path "../device_write/.." although other file paths are possible as well.

Note that shared filesystem 230' could also be presented to device operating system 101h after configuration process 231 as a separate drive or mounted filesystem for device 101. Device write directory 191x could also be referred to as RU read directory 191x. Or, directory 191x could appear in operating system 101h as a new directory 191x in existing filesystem 229 for device 101 after a configuration step 231. In addition, any files recorded in shared filesystem 230' may preferably be recorded in a nonvolatile state, such that the files remain available to device 101 and SRU 109 and RU 109b when (i) power is removed from device 101 and/or SRU 109 and RU 109b, and (ii) power is subsequently provided to device 101 and/or SRU 109 and RU 109b.

Device write directory 191x could include a plurality of different files 191y over the lifetime of operation of SRU 109 and device 101. Although a plurality of different files 191y are depicted in Figure 3a, not all files may be present in directory 191x within shared filesystem 230' at the same time and the different files depicted within device write directory 191x in Figure 3a could be written by device 101 and read by RU 109b at different points in time. In some exemplary embodiments, such as those depicted in Figure 3a, directory 191x could be initialized after a configuration process 231 with an empty set of files, where SRU 109 and RU 109b can wait for device 101 to send files, data, or commands or data for RU 109b within SRU 109 in the form of at least one file 191y.

Exemplary files for a device write directory 191x within device write and RU read memory 191 can comprise files 191y. Exemplary files 191y can comprise files for radio_parameters_input 191a, http(s)_request 191b, MO_sms 191c, FTP_file_upload 191d, command_script 191g, power_control_settings 191f, device_identity 101a, cryptographic_settings 191e, and date-time-set 191h. Other files or data for a device write directory 191x are possible as well without departing from the scope of the present disclosure. The files 191y can comprise text, numeric, or binary data according to encoding rules such as using URL encoding with http(s)_request, and UTF-8 for MO_sms 191c, and other possibilities exist as well. Device reporting libraries 101x in device 101 and radio unit read library 243 could agree or be configured to use a mutually compatible encoding rules for each of the files 191y and/or file types in directory 191x.

Radio_parameters_input 191a in files 191y can comprise data in a file format for radio settings 191a depicted in Figure 2b. Values could specify frequencies for RU 109b to utilize, RF networking technology to utilize, such as NB-IoT 104 or LTE Cat M1 or other settings as well. Radio_parameters_input 191a could specify a server B 110a name or address to utilize for the transmission of application data 112a through radio unit 109b, wireless network 102, and SRU-SP 106. Radio_parameters_input 191a could also include values for timers and counters for radio unit 109b to utilize, such as a retry timers and counters for the transmission of application data 112a. For embodiments where SRU 109 includes more than one eUICC 113, radio_parameters_input 191a can specify a preferred or primary eUICC 113 for RU 109b to utilize, where each different eUICC 113 could record and operate with a different operational profile 113d. In this manner, device 101 could select a set of network access credentials and network parameters for RU 109b.

Http(s)_request 191b, MO_sms 191c, and FTP_file_upload 191d could comprise the file format for stored data in a shared filesystem interface 230' equivalent to the values and data depicted and described for device write and radio unit read memory 191 in Figure 2b. Device 101 could write other application data 112a to device write directory and device write files 191y as well, including an outbound email message, a Message Queuing Telemetry Transport (MQTT) message, a Session Initiation Protocol (SIP) message, and other possibilities exist as well without departing from the scope of the present disclosure. Http(s)_request 191b, MO_sms 191c, and FTP_file_upload 191d and the other listed exemplary messages from the previous sentence individually or together could comprise exemplary application data 112a from Figure 1a, as depicted in Figure 3a.

Other possibilities exist as well for application data 112a in a file 191y in directory 191x for shared filesystem interface 230' without departing from the scope of the present disclosure. Application data 112a could also be written by device 101 in an encrypted form and/or comprise data for RU 109b to send to server B 110a as TCP or UDP datagrams. Files in application data 112a could include destination TCP or UDP port numbers to utilize for RU 109b to transmit the data to wireless network 102. Although single values of the files 191y for application data 112a are depicted in Figure 3a, a directory 191x could record multiple different values of the depicted files (such as multiple http(s)_request 191b files 191y for RU 109b to transmit or multiple different MO_sms 191c for RU 109b to transmit, etc.)

Command_script 191g can comprise a file with a set of sequential or programmatic commands for RU 109b to conduct, and could be equivalent or similar to an AT command script. Command script 191g could instruct RU 109b to periodically attach to wireless network 102 and query server B 110a (possibly through SRU-SP 106) in order to check for or obtain new server B data 112b for device 101. Command_script 191g could instruct RU 109b for actions for RU 109b to take when no connectivity to a primary or preferred wireless network 102 is available, including the steps to search for backup wireless networks 102. Command script 191g could also instruct RU 109 to periodically attach to wireless network 102 and query a different server such as server A 108.

In some exemplary embodiments, command_script 191g can comprise a traditional AT command script for RU 109b to execute. AT command script could use instructions from standards such as 3GPP TS 27.007 or TS 27.005, as well as related or subsequent versions of these standards. In some exemplary embodiments, the command_script 191g could include application data 112a within the script 191g, such that a single file or sequence of data can be passed to RU 109b, where the single file or sequence of data includes commands to attach to wireless network 102 and transmit the application data 112a for the command script. Or, the command_script 191g could refer to application data 112a in the form of a file 191y for RU 109b to transmit.

In another exemplary embodiment, processor 109d could read command_script 191g recorded by device 101 and write the command_script 191g to RU 109b using an internal bus 212, where bus 212 could operate with a traditional UART, serial, or SPI interface for RU 109b. In this manner, (i) commercially available radio units or modules with AT command interfaces could be utilized with an RU 109b while (ii) obtaining the benefits of a shared memory interface 230 to transfer the data from CPU 101b in device 101 and RU 109b. Processor 109d could also read application data 112a from a file 191y and pass or write the data to RU 109b. Or, RU 109b

could “fetch” or read the application data 112a in the form of a file 191y based on a file path passed along in the command_script 191g. A benefit of using command_script 191g from nonvolatile memory 109h is that CPU 101b can be in a sleep state when RU 109b reads the data and performs the operations specified by the command_script 191g. Other possibilities and benefits exist as well for a directory 191x in a shared filesystem interface 230’ to record a command_script 191g for RU 109b without departing from the scope of the present disclosure.

In some exemplary embodiments, command_script 191g could be recorded in a format other than a file 191y within memory 191, such as a sequence of binary data with a start and end memory address within memory 109g, and processor 109d could use libraries 243 and 246 to track the start and end memory addresses. As depicted and described in connection with Figure 2b, data in memory 109h such as device write memory 191 and RU write memory 192 can be recorded without using a file system. Other data depicted in Figure 3a could be recorded in memory 191 without using or requiring a traditional file system 229 for memory 191 and memory 192.

Power_control_settings 191f as a file 191y can include settings or values for use with a power control unit 109e by SRU 109 or RU 109b. Power_control_settings 191f could specify the time for RU 109b to remain in an active state or a connected state with wireless network 102 after transmitting application data 112a in order to receive a server B response 112b, such as an exemplary 5 seconds or an exemplary 10 seconds. Power_control_settings 191f could specify the use and values for a power saving mode (PSM) used by radio unit 109b, as well as a frequency or time for radio unit 109b to utilize a discontinuous receive (DRX) mode associated with wireless network 102. Power_control_settings 191g could specify a maximum power utilization by both SRU 109 in total and RU 109b. Power_control_settings 191f could specify the maximum RF power output for RU 109b to utilize in dBm or watts. In another exemplary embodiment, power_control_settings 191f can provide settings or commands for the operation of power control unit 109e.

Device_identity 101a written by a device 101 as a file 191y can comprise the identity of the device 101, which could be sent or transmitted by SRU 109 and RU 109b to SRU-SP 106 and/or server B 110a. Although not depicted in Figure 3a, a file 191y with a digital signature for device 101 could be written to device write directory 191x as well as a file 191y. The digital signature could be over a file or data in application data 112a for transmission to SRU-SP 106 and/or server B 110a through wireless network 102, and SRU-SP 106 or server B 110a could verify the digital signature for the device using a certificate for device 101 recorded in SRU-SP 106 or server B 110a, where the digital signature could be used by the receiving party of

application data 112a along with receipt of device_identity 101a to select the certificate for device 101 and verify the digital signature created by device 101. In this manner, SRU-SP 106 or server B 110a could verify that the application data 112a received by SRU-SP 106 or server B 110a was authentically originated by device 101.

5 Cryptographic_settings 191e can comprise values, parameters, public keys, and symmetric ciphering keys for RU 109b to utilize. Cryptographic_settings 191e can also be referred to as cryptographic parameters 191e. In an exemplary embodiment, cryptographic_settings 191e could include a public key or certificate for device 101, where device 101 continues to record and operate with the corresponding private key.
10 Cryptographic_settings 191e could specify parameters or values for SRU 109 and/or RU 109b to use with cryptographic algorithms 109q as depicted and described in connection with Figure 1c above. Cryptographic_settings 191e could select algorithms and/or key lengths to use with an ECDH key exchange, an ECC named curve such as, but not limited to, P-256, secp256k1, P-384, curve25519, and other possibilities exist as well, including algorithms supporting post-
15 quantum cryptography.

Cryptographic_settings 191e could specify which of a plurality of different possible public keys for device 101 recorded in memory 109h that SRU 109 or RU 109b should utilize with cryptographic_algorithms 109q. Cryptographic_settings 191e could also specify which of a plurality of possible secure hash algorithms SRU 109 or RU 109b should utilize in
20 communications over wireless network 102, such as, but not limited to SHA-256, SHA-384, RIPE-160, SHA-3, etc. Cryptographic_settings 191e as a file 191y could also select the use of a symmetric ciphering algorithm for SRU 109 or RU 109b to utilize with communications over wireless network 102, such as, but not limited to, AES-128, 3DES, AES-192, AES-256, blowfish, as well as specifying a cipher mode such as AES-CBC, AES-CTR, AES-SIV, etc.

25 A date-time-set file 191h could specify the current date and time for SRU 109. In exemplary embodiments, device 101 can record a clock 101m which continues to keep time when device 101 is powered down or off, such as a system time for a personal computer that continues to increment even when the personal computer is powered off (although a battery such as battery 101n could continue to provide a lower level of power in order for device 101 to
30 continue to keep track of time). Since SRU 109 may also be powered down when device 101 is powered down or off, SRU 109 may need an updated value for date and time when SRU 109 is powered up. The value for date and time for SRU 109 and/or RU 109b to utilize can be set by a file 191y with the current date and time. SRU 109 and RU 109a can utilize the date and time from a date-time-set 191h (along with an internal clock for SRU 109 from clock 207 or clock

205) to create timestamp 304a with files 191y recorded in directory 191x. In another exemplary embodiment, RU 109b can obtain the current date and time from wireless network 102, and utilize that date and time with RF clock 205 in order to continuously monitor and track time when SRU 109 is powered. In exemplary embodiments, (i) device 101 writes a value for date and time 191h with application data 112a before CPU 101b sleeps, then (ii) SRU 109 reads the date and time 191h and keeps time using an internal clock such as 205, and then (iii) SRU 109 uses the subsequent date and time calculated from the internal clock to write the date and time values associated with files written by SRU 109 in directory 192x (shown below in Figure 3b).

File system interface 230 in shared memory 109h can also include file parameters or attributes. File parameters may be specific to a selected file system driver 242 and selected file system 229 for directory 191x in Figure 3a and 192x below in Figure 3b, and the depicted file parameters in Figure 3a are exemplary. File parameters could include permissions 301a, which include values to specify permissions for users 302a and 302b with the file, such as read, write, execute, etc. Different permissions in permissions 301a could be associated with different users 302a and 302b. Users 302a and 302b could comprise users for directory 191x, such as a device OS and radio unit OS. Depicted users for users 302a and 302b are exemplary for Figure 3a. Note that the exemplary files 191y in directory 191x depict two users 302a and 302b comprising a first user of device 101 (identified as the OS 101h) and a second user of SRU 109 (identified as the OS for RU 109b). Other possibilities exist as well without departing from the scope of the present disclosure. The user 302a could be identified and selected from a user for files in memory 109g and filesystem 229.

File parameters for file system interface 230 can include a file size 303a, which can comprise the number of bytes for the recorded file 191y. Timestamp 304a can comprise the date and time when the file 191y was written in directory 191x. Timestamp 304a can be written by OS 101h. Although only a year is depicted in Figure 3a for some files, a time value in hours, minutes, and seconds could also be recorded. File parameters or attributes can also include the file name 305a. The file names 305a and file structure (e.g. encoding, XML fields, JSON fields, etc.) within files 191y and directory 191x could be pre-agreed via a specification mutually shared between device 101 and SRU 109 before device 101 begins operating with SRU 109. In addition, although device write directory 191x and RU write directory 192x are depicted in this Figure 3a and Figure 3b below as separate directories within shared filesystem 230', the two directories could be combined, and device write files 191y and RU write files 192y (below in Figure 3b) could be combined into the same, single directory within shared filesystem 230'.

Although the use of a file system interface 230 is depicted in Figure 3a as the structure of data written by device 101 and read by RU 109b, in some exemplary embodiments the use of a shared filesystem interface 230' and files 191y and directory 191x could be omitted and replaced by the use of a range of memory addresses in shared nonvolatile memory 109h for the data. In other words, a range or set of values for memory addresses in memory 109h could be used for data in each of the depicted files (instead of recording a file in a filesystem 230) such as a first memory address for the start of 191a data, a second memory address for the start of 191b data, etc. For these embodiments, the memory addresses for data elements (e.g. 191a, 191b, etc.) in shared memory 109h could be recorded in each of SRU firmware 109u, RU firmware 222, and device reporting libraries 101x. These embodiments without (A) the use of a shared filesystem interface 230' but with (B) shared nonvolatile memory 109h can obtain the benefits described herein, such as benefits from device 101 and RU 109b transferring data between the nodes using shared nonvolatile memory 109h. In other words, the use of a shared filesystem interface 230' can comprise an exemplary shared memory interface 230 of shared nonvolatile memory 109h, and other embodiments could use different structures of a shared nonvolatile memory 109h without departing from the scope of the present disclosure. A shared memory interface 230 recording information in shared memory 109h could include a shared filesystem interface.

As described herein and as a need in the art, the use of shared nonvolatile memory 109h allows device 101 and RU 109b to transfer data between the components using independent sleep states. Thus, both CPU 101b and RU 109b do not both need to be in active states in order to transfer data (thereby allowing overall power consumption for a device 101 to be conserved). In contrast, with conventional technology such as with traditional AT commands for radio units such as traditional cellular modems in devices, both the radio unit and the primary CPU for a device both have to be active (e.g. not in sleep states) in order for AT commands and responses to be transferred between the CPU and the radio unit. Another benefit is that using a shared memory interface in nonvolatile memory (which could be a shared filesystem interface 230) allows both device 101 and SRU 109 to read the last state. Processors with conventional technology could (a) enter deep sleep or off states where RAM memory is flushed, and (b) not track the status of the last commands written or data transferred between device 101 and RU 109b, and thus (c) require a longer restart process after exiting an off or deep sleep state, thereby consuming more power or energy. Conventional technology may require start of an AT command script upon each resumption from an off state for device 101 or SRU 109. Using a

shared nonvolatile memory interface 230, intermediate states at the last power down can be both read and quickly resumed.

Figure 3b

Figure 3b is a graphical illustration of a shared filesystem interface for a radio storage unit and a device, using a radio unit write directory, in accordance with exemplary embodiments. SRU 109 can operate and record a RU write and device read memory 192, where memory 192 can reside within shared nonvolatile memory 109h. Shared nonvolatile memory 109h could be formatted with a shared filesystem 230' using the configuration steps 231, such that (i) device 101 could read from shared filesystem 230' and (ii) RU 109u and SRU 109 could write data to shared filesystem 230'. In addition, memory 192 can be referred to as "radio unit write and device read memory" 192 or simply memory 192 or RU write memory 192 or device read memory 192.

Shared filesystem interface 230 RU write directory 192x is depicted as viewable and readable from device operating system 101h with a file path ".../radio-unit_write/..." although other file paths are possible as well. Note that shared filesystem 230' could also be presented to device operating system 101h after configuration process 231 as a separate drive or mounted filesystem for device 101. Further, a configuration process 231 could insert directory 192x into filesystem 229, such that directory 192x appears as a separate directory within device memory 109g. Device read directory 192x could also be referred to as RU write directory 192x. In addition, any files recorded in shared filesystem 230' may preferably be recorded in a nonvolatile state, such that the files remain available to device 101 and SRU 109 and RU 109b when power is removed from device 101 and/or SRU 109 and RU 109b. In addition, memory 109h with files 192y and 191y could continue to record files 192y and files 191y when either device 101 or SRU 109 or RU 109b enter a deep sleep mode or state. The deep sleep mode or state could comprise device RAM 101e and SRU RAM 213 (shown in Figure 2a) being flushed or cleared, such that the RAM no longer records usable data. If SRU RAM 213 comprises NVRAM, then SRU RAM 213 would not normally be cleared when power is removed from SRU 109.

RU write directory 192x could include a plurality of different files 192y over the lifetime of operation of SRU 109 and device 101. Although a plurality of different files 192y are depicted in Figure 3b, (a) not all files may be present in directory 192x within shared filesystem 230' at the same time and (b) the different files depicted within device write directory 192x could be written by RU 109b and read by device 101 at different points in time. In some

exemplary embodiments, such as those depicted in Figure 3b, directory 192x could be initialized after a configuration process 231 with a preliminary set of files, where device 101 could read files such as, but not limited to, radio_parameters_loaded 192a, log_file 306, and error_log 309 both (i) after boot or exit from a sleep state of device 101 or CPU 101b, and (ii) before RU 109b is powered by power control unit 109e after SRU 109 receives power from pins 109x.

Exemplary files for a RU write directory 192x within RU write and device read memory 192 can comprise files 192y. Exemplary files 192y can comprise files for radio_parameters_loaded 192a, http(s)_response 192b, MT_sms 192c (e.g. mobile terminated SMS), FTP_file_downloaded 192d, log_file 306, server B digital signature 307, SRU_identity 109i, cryptographic_parameters 192e, and radio_network_status 308, error_log 309, and network-date-time 309b. Other files or data for a device write directory 192x are possible as well without departing from the scope of the present disclosure. The files 192y can comprise text, numeric, or binary data according to encoding rules such as using UTF-8, or ISO-8859-X encoding with http(s)_response 192b, and UTF-8 for MT_sms 192c, and other possibilities exist as well. Device reporting libraries 101x in device 101 and radio unit write library 246 could agree or be configured to use a mutually compatible encoding rules for each of the files 192y and/or file types in directory 192x.

Radio_parameters_loaded 192a in files 192y can comprise data in a file format for radio settings 192a (or "radio configuration") depicted in Figure 2b. Radio_parameters_loaded 192a could comprise the default values or last utilized parameters or loaded parameters for radio 109b before device 101 writes data to radio_parameters_settings 191a. Values could specify frequencies, bands, or channels that RU 109b is configured for or available to RU 109b, RF networking technology to utilize, such as NB-IoT 104, a WiFi version such as n, ac, ah, or ax, or LTE Cat M1 or other settings as well. Radio_parameters_loaded 192a could identify a server B 110a name or address that SRU 109 or RU 109b utilizes for the transmission of application data 112a through radio unit 109b, wireless network 102, and SRU-SP 106. Radio_parameters_loaded 192a could also include values for timers and counters that radio unit 109b implements or utilizes, such as a retry timers and counters for the transmission of application data 112a and receipt of server B response 112b data. For embodiments where SRU 109 includes more than one eUICC 113, radio_parameters_loaded 192a can identify the preferred or primary eUICC 113 which RU 109b utilizes (such as with eUICC ID 113a), where each different eUICC 113 could record and operated with a different operational profile 113d. In this manner, device 101 could identify which set of network access credentials and network parameters are used by RU 109b with a wireless network 102.

Http(s)_response 192b, MT_sms 192c, and FTP_file_downloaded 192d could comprise the file format for stored data in a shared filesystem interface 230' equivalent to the values and data depicted and described for RU write and device read memory 192 in Figure 2b. RU 109b could write other data or server B response 112b data to RU write directory 192x and RU write files 192y as well, including an inbound email message, a Message Queuing Telemetry Transport (MQTT) response, a Session Initiation Protocol (SIP) message received through wireless network 102, and other possibilities exist as well without departing from the scope of the present disclosure. Http(s)_response 192b, MT_sms 192c, and FTP_file_downloaded 192d individually (and/or exemplary messages from the previous sentence) or together could comprise exemplary server B response 112b data from Figure 1a, as depicted in Figure 3b.

Other possibilities exist as well for server B response 112b data in a file 192y in directory 192x for shared filesystem interface 230' without departing from the scope of the present disclosure. Server B response data 112b could also be written by RU 109b in an encrypted form and/or comprise data for RU 109b to send to device 101, where device 101 decrypts the server B response 112b data. Although single values of the files 192y for server B response data 112b are depicted in Figure 3b, a directory 192x could record multiple different values of the depicted files (such as multiple http(s)_request 191b files 191x for RU 109b to transmit or multiple different MO_sms 191c for RU 109b to transmit).

Log_file 306 in files 192x can comprise a log file of actions or steps performed by RU 109b or SRU 109. A log_file 306 logging level such as a debug level, an "info" level, or a warning or notice level could be specified in radio_parameters_input 191a. Log_file 306 could also comprise a sub-directory instead of a file 192x. Further, any of the exemplary file data depicted in Figure 3b or Figure 3a could comprise directory names for sub-directories in directory 192x or 191x. Other possibilities exist as well for a directory 192x in a shared filesystem interface 230' to record at least one log_file 306 written by SRU 109 or RU 109b and read by device 101 without departing from the scope of the present disclosure.

Server B digital signature 307 from server B 110a could be written to RU write directory 192x as well. The digital signature could be over a file or data in server B response data 112b received by RU 109b via wireless network 102 from SRU-SP 106 and originated by server B 110a. Device 101 could verify the digital signature for the server B using a certificate recorded in device 101 for server B 110a, such as device 101 recording a certificate 110c, which is depicted and described in connection with Figure 1a. In this manner, device 101 could verify that the server B response data 112b received by RU 109b and passed to device 101 was authentically originated by server B 110a.

In some exemplary embodiments, the use of a server B digital signature 307 could be omitted, and device 101 could rely upon secure communications between RU 109b and server A 108 (such as via a mutually derived symmetric ciphering key 223a depicted in Figure 4b below). For these exemplary embodiments, if device 101 trusts server A 108, then device 101 may trust data that server A 108 receives from server B 110a (e.g. device 101 can rely on (x) a mutually authenticated TLS 114 session and (y) RU 109b security such as key 223a, where the use of (x) and (y) could also be specified in radio_parameters_input 191a). In other exemplary embodiments, a server B digital signature 307 could be included with server B response 112b data, and thus a separate file 192y for server digital signature 307 may not be recorded in a directory 192x.

Power_control_settings 192g as a file 192y can include settings or values used by a power control unit 109e for SRU 109 or RU 109b. Power_control_setting 192g as a file 192y can represent the current state or mode for power control unit 109e operating in SRU 109. In this manner, device 101 could read file power_control_settings 192g in order to determine the current settings, configuration, or values used by a power control unit 109e. Power_control_settings 192g can comprise the running configuration of power control unit 109e and comprise data equivalent to that depicted and described for power_control_settings 191g in Figure 3a above. In other words, power_control_settings 191g can comprise values input by device 101, and power_control_settings 191g can comprise the loaded and operating configuration of power control unit 109e, which could be read by device 101 as a file 192y. In exemplary embodiments, RU 109b and/or SRU 109 writes both power_control_settings 192g and radio_parameters_loaded 192a to directory 192x upon boot of SRU 109 and RU 109b.

Cryptographic_parameters 192e can comprise values, parameters, public keys, and symmetric ciphering keys that RU 109b utilizes. In an exemplary embodiment, cryptographic_parameters 192e could include a public key or certificate for SRU 109 such as a public key SRU PK 101w for SRU private key 109p, where SRU 109 continues to record and operate with the corresponding private key 109p. Cryptographic_parameters 192e could specify parameters or values utilized by SRU 109 and/or RU 109b with cryptographic algorithms 109q as depicted and described in connection with Figure 1c above and also parameters 191e in Figure 3a above. Cryptographic_parameters 192e could comprise settings, parameters, values, and configuration data equivalent to those described for cryptographic_settings 191e, where (a) cryptographic_parameters 192e could include a list of available options for SRU 109 and RU 109b and a selected list of implemented or running options configured for SRU 109 and RU 109b, and (b) cryptographic_settings 191e could comprise the selection or input by device 101

of particular values, settings, or algorithms as described above in Figure 3a. In other words, parameters 191e could comprise a selected subset of parameters 192e, where 192e is a list of all options, and 191e is a selected list chosen by device 101 or SRU 109.

Radio_network_status 308 can comprise the current status of radio unit 109b with
5 wireless network 102. For example, radio_network_status 308 could include the radio state such as “radio unit off”, “attached”, “transmitting”, or “idle”, “access denied”, “credentials invalid”, and other possibilities exist as well. Radio_network_status 308 could also comprise a state or operating mode used by RU 109b. In exemplary embodiments, radio_network_status 308 can also include a list of all available wireless networks observed by radio unit 109b using the
10 current settings specified in radio_parameters_input 191a. Radio_network_status 308 could also record power utilization or settings for RF front end 203, such as current or last transmit power in dBm, such as 0 dBm, 23 dBm, and other possibilities exist as well. Error_log 309 can comprise a running list of operating errors incurred by SRU 109 and/or RU 109b, such as recording that wireless network 102 is not available, SRU-SP 106 does not receive a response
15 from server B 110a, that authentication or encryption using cryptographic_parameters 192x fails, an out of memory state for memory 109h, improperly formatted or corrupted files for files 191x or 192x, and other possibilities exist as well for data recorded in an error_log 309.

Although not depicted in Figure 3b, files 192x in RU write directory 192x could include a file for current or last obtained geographical position or coordinates as measured or determined
20 by RU 109b. In exemplary embodiments, RU 109b can include a Global Positioning System (GPS) receiver or Global Navigation Satellite System (GNSS) receiver, and obtain geographical coordinates or position. The geographical coordinates could be written to a file 192y in RU write directory 192x, which could subsequently be read by device 101.

File system interface 230 in shared memory 109h for RU write directory 192x can also
25 include file parameters or attributes. File parameters may be specific to a selected file system driver 242 and selected file system for directory 191x and 192x below, and the depicted file parameters in Figure 3a are exemplary. File parameters depicted in Figure 3b could be equivalent to file parameters depicted and described above with Figure 3a. File permissions 301b can be for file permissions in RU write directory 192x. Users 302b can be the users
30 identified for files 192y in directory 192x, and users 302b can be the same or equivalent to users 302a in directory 191x. Files 192y can have file names 305b similar to file names 305a in directory 191x. The file names 305b and file structure (e.g. encoding, XML fields, JSON fields, etc.) within files 192y could be pre-agreed via a specification mutually shared between device 101 and SRU 109 before device 101 begins operating with SRU 109. As one example, data

reporting libraries 101x in device 101 could record a structure 192x' for directory 192x (depicted in Figure 1b), such that data reporting libraries 101x could read data from directory 192x.

Although the use of a file system interface 230 is depicted in Figure 3b as the structure of data written by RU 109b and read by device 101, in some exemplary embodiments the use of a shared filesystem interface 230' and files 192y and directory 192x could be omitted and replaced by the use of a range of memory addresses in shared nonvolatile memory 109h for the data. In other words, a range or set of values for memory addresses in memory 109h could be used for data in each of the depicted files (instead of recording a file in a filesystem 230) such as a first memory address for the start of 192a data, a second memory address for the start of 192b data, etc. For these embodiments, the memory addresses for data elements (e.g. 192a, 192b, etc.) could be recorded in both SRU firmware 109u and device reporting libraries 101x. These embodiments without (A) the use of a shared filesystem interface 230' but with (B) shared nonvolatile memory 109h can obtain the benefits described herein, such as from device 101 and RU 109b transferring data between the nodes using shared nonvolatile memory 109h. Embodiments without shared filesystem interface 230' are also described above in Figure 3a as well as benefits generally from using a shared nonvolatile memory 109h without requiring the use of filesystem 230.

Figure 3c

Figure 3c is a flow chart illustrating exemplary steps for a device and a storage radio unit to utilize a shared filesystem interface, in accordance with exemplary embodiments. Before SRU 109, RU 109b, and device 101 begin utilizing the shared filesystem interface 230', SRU 109 could conduct the configuration process 231 depicted and described above in connection with Figure 2b and also Figure 2f. Upon the conclusion of a configuration process 231, device write directory 191x could contain an empty 310 set of files for both (a) application data 112a in a device write directory 191x and (b) server B response 112b data in a RU write directory 192x. Although not depicted in Figure 3c, in exemplary embodiments, upon conclusion of a configuration process 231, other data not depicted in Figure 3c could be present in directory 191x and 192x, such as (X) device 101 writing to 191x files 191y of date-time-set 191g, device_identity 101i, radio_parameters_input 191a, and power_control_settings 191g, and (Y) RU 109b writing to 192x files for 192y of radio_parameters_loaded 192a, log_file 306, power_control_settings 192g, cryptographic_parameters 192e, and radio_network_status 308.

At step 311, shared filesystem interface 230' could be recorded in shared nonvolatile memory 109h and initialized with directory 191x ready for input of application data 112a from device 101. At step 311, directory 191x can begin without files or data recorded in directory 191x for application data 112a (although other data could be recorded in files 191y in directory 191x by device 101, such as the exemplary files described in the paragraph above related to the operation and configuration of a SRU 109 with RU 109b.) At step 312, shared filesystem interface 230' can be initialized with directory 192x ready for input of server B response data 112b from RU 109b. At step 312, directory 192x can begin without files or data recorded in directory 192x for server B response 112b data (although other data could be recorded in files 192y in directory 192x by SRU 109 and RU 109b, such as the exemplary files described in the paragraph above related to the operation and configuration of a SRU 109 with RU 109b.)

At step 313, device 101 could obtain and record data for server B 110a. The data could be obtained from a transducer for device 101 such as a sensor. Or, device 101 could prefer to conduct periodic reporting or registration with server B 110a without sensor data. In either case, device 101 could structure a message for server B 110a in the format of an HTTP or HTTPS request or URL. The data for server B 110a could also comprise an XML file or other text or binary data. The URL could include parameters such as a query string or values associated with the sensor or device 101. Device 101 could write the URL in the form of an http_request 191b file 191y to the shared filesystem interface in directory 191x. The http_request 191b could comprise application data 112a for server B 110a. An exemplary file 191y for http_request 191b is also depicted in Figure 3c. The shared filesystem 230' can also record file parameters, such as permissions and a timestamp, as well as a file name 305a. Although not depicted in Figure 3c, device 101 could also include a digital signature for http_request 191b, where the signature was created with private key for the device 101. Server B 110a could record the corresponding device public key. Or, in exemplary embodiments a digital signature for device 101 could be included within the application data 112a. In exemplary embodiments, application data 112a could be text or binary data and does not need to be in the format of an HTTP, HTTPS, or XML request or use a URL.

At step 314, power control unit 109e could power up RU 109b from a sleep state to an active state. SRU 109 could operate either RU polling 232 or use a trigger library 245 to detect that application data 112a in the form of http_request 191b has been written as a file 191y to directory 191x. RU 109b could subsequently use read library 243 from SRU firmware 191u to read http_request 191b from directory 191x. RU 109b can attach to wireless network 102, such as using an NB-IoT 104 attach procedure, and other wireless networking technologies could be

utilized as well, such as LTE CAT M, LTE Advanced, 5G, WiFi network attachment and authentication, etc.

RU 109b can subsequently transmit http_request 191b to SRU-SP 106 or directly to server B 110a through wireless network 102. The selection of a destination for transmission of http_request 191b could be specified in 191b or possibly parameters 191a. In exemplary
5 embodiments, RU 109b transmits http_request 191b and other application data 112a to server A 108, which is depicted in Figure 3c and Figure 4a. Upon successful transmission of http_request 191b to server A 108 (or server B 110a if RU 109b communicates with server B 110a) through wireless network 102, then upon conclusion of a step 314 then RU 109b can (i) delete the file
10 191y for http_request 191b, or (ii) move file 191y with http_request 191b to a log_file 306 or associated sub-directory for logged or historical data.

Continuing at step 314, the deletion of the file from directory 191x by RU 109b can signal or indicate to device 101 that the http_request 191b data has been successfully transmitted to wireless network 102. Device 101 can also read device write directory 191x (as shown by the
15 exemplary file permissions 301a from Figure 3a above). Note that a shared filesystem interface 230' provides benefits for both device 101 and RU 109b, since device 101 could enter a sleep state and not remain actively connected to SRU 109 (other than providing power through pins 109x). Device 101 could subsequently wake at a later time (such as returning from a sleep state to an active state) and observe that the file 191y for http_request 191b has been deleted, thereby
20 indicating successful transmission of the data through wireless network 102. Device 101 could also record the outgoing http_request 191b in other memory within device 101, if device 101 needs access to the transmitted data, so deletion of the file 191y for http_request 191b does not prevent device 101 from continuing to record the data until a full reporting cycle such as http_response 192b is read by device 101. For example, if device 101 does not receive
25 http_response 192b in an expected period of time, such as an exemplary 5 seconds, then device 101 could subsequently write http_request 191b to directory 191x a second time after RU 109b deletes the file.

Further, although the deletion of files in shared filesystem 230' is depicted in Figure 3c, such as with a step 314 and also 316 below, in some embodiments the corresponding files 191y
30 and 192y can optionally not be deleted and remain recorded in shared filesystem 230'. For these embodiments, device 101 and RU 109b can each detect the presence of new files for processing based on updated timestamps 304 for the files 191y and 192y, and other possibilities exist as well for continuing to record data in shared filesystem 230' after the data or files have been transmitted and received.

At step 315, RU 109b can receive an http_response 192b via wireless network 102. Http_response 192b could be generated by server B 110a based on http_request 191b and passed to server A 108. Http_response 192b can comprise a file with an *.html or *.htm extension, and other possibilities exist as well. Although depicted as an http_response 192b, the response 112b
5 could be text or binary data without formatting or structure according to the http protocol. Note that http_response 192b can comprise server B response 112b data, as depicted in Figure 3c. RU 109b can subsequently use SRU write libraries 246 in order to write the received http_response 192b to a file 192y in directory 192x. Http_response 192b as a file 192y can include file attributes, such as (i) a timestamp for when the file is written by RU 109b and (ii) a
10 file size in bytes. SRU 109 could use an internal clock 205 to keep time for writing the timestamp, where the internal clock 205 for SRU 109 could be initially synchronized with device 101 using a date and time input/set 191g from device 101 through files 191y.

As depicted in Figure 3c and continuing with step 315, RU 109b could also receive a server B digital signature 307 for the http_response 192b, where digital signature 307 is over
15 http_response 192b data and created using server B private key corresponding to the server B public key in a certificate 110c. Device 101 can record certificate 110c and use the public key from the certificate in order to verify digital signature 308. In some exemplary embodiments, both http_request 191b and http_response 192b can be communicated by RU 109b and also recorded by shared filesystem 230' in an encrypted form, where device 101 and server B 110a
20 record and operate with a shared secret ciphering key. In that and related embodiments, then a server digital signature 307 could be optionally omitted (since only both device 101 and server B 110a could mutually encrypt and decrypt data 112a and 112b by commonly sharing the same shared secret ciphering key). In some embodiments, the use of a digital signature 307 for server B response 112b could be omitted.

At step 315, device 101 could read http_response 192b and server B digital signature 307
25 from files 192y in directory 192x in shared filesystem 230'. Device 101 can subsequently process the data in http_response 192b, which could comprise a confirmation that http_request 191b was properly received by server B 110b. Http_response 192b could also include commands or instructions for device 101, such as a value for an actuator 109y or a setting for
30 device 101 to implement or utilize. As depicted in Figure 3c, device 101 could also delete the file 192y for http_response 192b after reading and processing the data in a step 316. In this manner, shared memory 109h can be conserved. Further, the deletion of http_response 192b from shared filesystem 230' can comprise a confirmation for SRU 109 and RU 109b that device 101 has successfully received the file or http_response 192b data.

Note that recording http_response 192b (or http_request 191b) in nonvolatile memory 109h provides benefits, such that both device 101 and RU 109b can enter and exit sleep states independently. In other words, both RU 109b for SRU 109 and CPU 101b for device 101 do not need to both be active at the same time in order to successfully transfer data between the two components. In this manner, a device 101 could better conserve energy stored in a battery 101n and extend the life of a charge for the battery 101n. With conventional technology that does not utilize shared nonvolatile memory 109h, both device 101 and RU 109b may need to be in active (or non sleep) states or modes in order to transfer data between device 101 and RU 109b.

Although an http_request 191b is depicted in Figure 3c, other application data 112a could be written by device 101, read by RU 109b, and transmitted by RU 109b through wireless network 102 as well, such as an SMS, FTP file, MQTT message, SIP message, CoAP message, or simply an XML or JSON file. Text or binary data could be written by device 101 as application data 112a as well. Likewise, although an http_response 192b is depicted in Figure 3c, other server B response 112b data could be written by RU 109b when received by RU 109b from wireless network 102, using the exemplary protocols or data structures from the previous sentence. Other possibilities exist as well for using a shared filesystem interface 230' in shared nonvolatile memory 109h operating in a SRU 109 with RU 109b without departing from the scope of the present disclosure.

Figure 4a

Figure 4a is a simplified message flow diagram illustrating an exemplary system with exemplary data sent and received by a device, a storage radio unit, a radio unit, a server A, and a server B, in accordance with exemplary embodiments. System 400 can include device 101, storage radio unit (SRU) 109, radio unit (RU) 109b, server A 108, and server B 110a. Device 101 can comprise a computing device such as depicted and described in connection with Figure 1a, Figure 1b and other figures above. Storage radio unit 109 can comprise a unit with a radio and nonvolatile memory 101a as depicted and described in Figure 1a, Figure 1c, Figure 2a, and other figures above. Although storage radio unit 109 is depicted in Figure 4a as a separate element than radio unit 109b, radio unit 109b can operate within storage radio unit 109. SRU 109 can communicate with device 101 via external bus 202 and electrical pins 109x. SRU 109 can communicate with RU 109b via internal bus 212.

For a system 400, server A 108 and server B 110a can comprise servers connected to an IP network 111 in order to receive data from RU 109b and device 101, where server A 108 and server B 110a are depicted and described in connection with Figure 1a and other figures above.

Server A 108 can communicate with RU 109b through both wireless network 102 and IP network 111. Server A 108 and server B 110a can communicate through IP network 111. The exemplary steps and message flows depicted in Figure 4a provide an exemplary illustrative summary for the transfer of application data 112a from device 101 to server 110b using SRU 109, RU 109b, and a server A 108 operating with SRU-SP 106.

At step 401a, device 101 can be manufactured, where the exemplary components for a device 101 from Figure 1b can be assembled and connected. At step 401b, device 101 can be configured, such as loading boot firmware and configuration data for device 101 in ROM 101c. Step 401b can include recording driver 101k such that device 101 can read data from a storage memory such as a SRU 109. Step 401b can also include device 101 recording drivers for a shared filesystem interface 230' as well as data reporting libraries 101x.

Storage radio unit 109 can undergo a step 260 as depicted and described in connection with Figure 2f, which can include manufacturing of SRU 109, recording boot and SRU firmware 109u within SRU 109, recording SRU ID 109i, SRU private key SRU SK 101p, cryptographic algorithms 109q and associated parameters 192e, a bootstrap profile 113c for an eUICC within SRU 109, and certificates for server A 108 and server B 110a. Other data can be recorded in a step 260 for an SRU 109 as well. Although step 260 is depicted for an SRU 109 in Figure 4a, an SRU-SP 106 could perform many of the steps such as recording the data in SRU 109 using a configuration unit equivalent to configuration unit 110d. Step 260 includes a step 253 where SRU 109 is inserted into device 101. In exemplary embodiments, step 260 can take place before step 401c and step 402 for device 101.

At step 401c, device 101 can power up or boot, load drivers for shared filesystem interface 230', and provide power to SRU 109 via electrical pins 109x. Device 101 can then be configured to read and write data with SRU 109. In exemplary embodiments, operating firmware for device 101 can be recorded in memory 109g in a filesystem 229 in SRU 109, and device 101 can load operating firmware for device 101 after SRU 109 boots in a step 402.

At step 402, SRU 109 can boot and use boot firmware 217 and boot configuration 218 to load SRU firmware 109u, where the components or executable code within SRU firmware 109u are depicted and described above in connection with Figure 2e. For embodiments where SRU 109 begins operating for the first time with device 101, SRU 109 can conduct a configuration process 231 for shared memory 109h, which is depicted and described above in connection with Figure 2b and Figure 2f. After a configuration process 231, a shared filesystem interface 230' in shared memory 109h can be available to RU 109b and device 101 for communication between the two nodes.

At step 403, device 101 can be operating in a regular manner, such as collecting data from sensors 101f and controlling actuators 101y according to data or instructions in a device program 101i. Step 403 can include device 101 conducting a mount process to attach SRU 109, in order to read shared memory 191 and 192 (which could be structured with a shared filesystem interface 230). Device 101 can obtain or calculate data for server B 110b in the form of application data 112a, which could comprise an http_request 191b. Although an http_request 191b is depicted in Figure 4a as application data 112a, other exemplary data could be transferred as well such as XML files, JSON files, a MQTT message, an SMS message, etc, which are also described in Figure 3a.

SRU 109 can then conduct steps 311 and 312 as depicted and described in connection with Figure 3c above, where shared filesystem interface 230' is ready for read and write operations by both device 101 and RU 109b. Processor 109d in SRU 109 can then begin running a trigger library 245 in order to wake RU 109b and signal RU 109b when data in directory 191x is available for RU 109b to read. In step 404, RU 109b can boot and then RU 109b can enter a radio off or deep sleep state. In exemplary embodiments, a power control unit 109e for SRU 109 keeps RU 109b in a sleep state or deep sleep state unless (i) there is application data 112a written to directory 191x or (ii) RU 109b is waiting for data or a server B response 112b after the transmission of application data 112a. A step 401d for server A 108 can comprise server A 108 recording data for SRU 109 and RU 109b, such as SRU identity 109i, and a public key or SRP PK 109w corresponding to SRU private key SRU SK 109p.

Device 101 can then conduct a step 313, where the http_request 191b from a step 402 is written to shared filesystem interface 230' in device write directory 191x. In exemplary embodiments, device 101 then conducts a step 429a to place CPU 101b and associated components from Figure 1b for device 101 in a sleep state. The sleep state in a step 429a can be continuous until device 101 receives a signal 429c below to wake. Or, device 101 could use a timer to sleep for a specified time in a step 429b, for at least an exemplary 5 seconds, where CPU 101b sleeps until the timer expires. Note that device 101 can continue to power SRU 109 via pins 109x when device 101 or CPU 109b enters a sleep state. Upon expiration of the timer for a step 429a, CPU 101b could begin a polling step 233 to query or check for updated files 192y, such as polling for a server B response 112b data written to RU write directory 192x.

In a step 405, the trigger library 245 that was loaded above in SRU 109 after step 312 can (A) detect new data for RU 109b to process and transmit, and consequently (B) power control unit 109e can transition RU 109b from a sleep or off state to an active state. In message 406a, SRU 109 can use AT control stack 109s to activate radio unit 109b and instruct RU 109b

to connect with a wireless network 102 and use a profile from eUICC 113 in the connection steps with wireless network 102. Message 406a could be transmitted through internal bus 212 in Figure 2a.

In some embodiments, radio unit 109b could comprise a traditional module or modem for wireless networks 102, and the radio unit 109b can be controlled by SRU 109 and processor 109d using AT control stack 109s (in Figure 1c) and messages such as message 406a. In other embodiments where radio unit 109b is integrated with SRU 109, then other commands and control besides an AT control stack 109s could be utilized by SRU 109 and processor 109d to control radio unit 109b. In exemplary embodiments, device 101 and SRU 109 could use a shared nonvolatile memory interface 230 to communicate between the components, and SRU 109 could use an AT control stack 109s to communicate internally over internal bus 212 between processor 109c and radio unit 109b. Or, SRU firmware 109u could include commands other than an AT control stack 109s to operate a radio unit 109b.

At step 406b, RU 109b can be powered up and receive message 406a with the instruction to connect with wireless network 102. At step 407a, RU 109b can (i) attach to wireless network 102 with an identity such as GUTI, TMSI, IMSI, SUPI, or 5G-GUTI from eUICC or operating profile 113d (or bootstrap profile 113c), and (ii) receive a challenge from wireless network 102. RU 109b can use a SIM card interface or commands with eUICC 113 in order to (i) read an identity such as IMSI or SUPI from the eUICC and (ii) send a challenge (or RAND or equivalent) received from wireless network 102 to the eUICC 113, and (iii) receive a response or SRES or equivalent from the eUICC 113 using an operating profile 113c to authenticate with wireless network 102.

At step 407b, RU 109b can transmit the SRES or equivalent to wireless network 102 and conclude the authentication process. At the conclusion of step 407b, RU 109b can enter a state with wireless network 102 such as a radio resource control (RRC) connected state (or equivalent for NB-IoT 104 networks or 5G networks) where RU 109b can transmit data to server A 108. Steps 407a and 407b could comprise the steps required for RU 109b to attach to a WiFi network as well, for embodiments where wireless network 102 comprises a network supporting the family of 802.11 standards.

At step 407c, RU 109b can read PKI keys 109v from radio unit only memory 109r, as well as cryptographic settings 191e (also referred to as cryptographic parameters 191e) from files 191y in directory 191x, and cryptographic algorithms 109q. At step 407d, RU 109b can read SRU identity 109i and a name or address for server A 108 from memory 109r. The name

or address for server A 108 can comprise the destination IP address for IP network 111 where RU 109b will transmit application data 112a.

RU 109b can then conduct a step 314, where step 314 was depicted and described in connection with Figure 3c above. RU 109b or SRU 109 can use read libraries 243 to read
5 http_request 191b from device write directory 191x in shared filesystem interface 230' using a step 408a. Although RU 109b is depicted in Figure 4a as reading http_request 191b from directory 191x, in an exemplary embodiment processor 109d in SRU 109 could (i) read http_request 191b (or other data in directory 191x) and (ii) pass the data through internal bus 212 to RU 109b, and (iii) instruct RU 109b to transmit the data, such as through using an AT
10 control stack 109s with RU 109b. In general, operations describing RU 109b as "reading" data such as files 191y in directory 191x can comprise processor 109d as reading the data and then processor 109d writing the data to RU 109b over an internal bus 212.

At step 409, RU 109b can derive (i) an ephemeral ECC private key SKe 409a (in Figure 4b below) using a named curve and associated parameters from cryptographic parameters 191e
15 and (ii) a corresponding ECC public key PKe 409b. In exemplary embodiments, the named curve and associated parameters from cryptographic parameters 191e are also utilized by an SRU-SP public key SRU-SP PK 106b recorded in memory 109r that corresponds to SRU-SP private key SRU-SP SK 106a. At step 409, RU 109b can conduct an ECDH key exchange 441 (in Figure 4b below) using at least the derived ECC private key SKe 409a in step 409 above and
20 the SRU-SP public key SRU-SP PK 106b in order to calculate a mutually derived shared secret key 223 with server A 108.

Note that the ECDH key exchange 441 in step 409 could also use as input the SRU private key SRU SK 109p in conjunction with the derived ephemeral ECC private key. In other words, the input to an ECDH key exchange 441 for RU 109b can include the sum of the two
25 private keys {SRU SK 109p and derived private key SKe 409a} mod n, where n is a curve parameter. For embodiments without use of SRU SK 109p in ECDH key exchange step 441, then the corresponding ECDH key exchange 442 (in Figure 4b below) can also omit the SRU PK 109w.

In some exemplary embodiments, the use of an ECDH key exchange 441 in step 409 can
30 be omitted, and RU 109b could use recorded symmetric key 127b depicted and described as a symmetric ciphering key instead of using a derived key 223. For these embodiments that omit use of an ECDH key exchange 441, the server A 108 could also record a symmetric key 127b for SRU 109 with SRU ID 109i and use the symmetric key 127b as a symmetric deciphering key. Details for an exemplary ECDH key exchange 441 in a step 409 are depicted and described

in connection with Figure 4b below. Encryption by RU 109b of application data 112a could be via symmetric key derived from symmetric key 127b. In other words, “symmetric key 127b” would be a master key that is not directly used for encryption, but a symmetric ciphering key for transmitting data with wireless network 102 could be derived using the “symmetric key 127b” as a master key.

At step 410, RU 109b can compress the http_request 191b data using a compression algorithm such as gzip and other possibilities exist as well. At step 411, RU 109b can encrypt (i) the compressed http_request 191b and (ii) device identity 101a (read from directory 191x in Figure 3a) with a symmetric ciphering algorithms such as AES and the mutually derived shared secret key 223 from a step 409 (or alternatively a recorded symmetric key 127b instead of mutually deriving the key 223). The output from a step 411 can comprise a ciphertext. RU 109b can then transmit a message 412 to server A 108, where message 412 can include SRU identity 109i, the derived ephemeral public key PKe 409a, and a ciphertext comprising encrypted values for 101a and 191b, where the ciphertext is encrypted with key 223a (and key 223a is depicted and described in connection with Figure 4b below).

For some exemplary embodiments, a recorded, pre-shared symmetric key 127b can be used with a DTLS-PSK session with server A 108 or with server B 110a, where application data 112a such as http_request 191b is encrypted with the symmetric key 127b (from Figure 1c) and a symmetric ciphering algorithm. For these embodiments, the generation of an ephemeral PKI key pair in a step 409 could be omitted. For embodiments which use a symmetric key 127b and server B 110b also records or stores the pre-shared symmetric key 127b, then (i) RU 109b could send the encrypted application data 112a directly to server B 110a and (ii) optionally bypass sending the application data 112a to server A 108. For these embodiments with the use of a pre-shared secret key 127b, and server B response 112b data received by RU 109b is decrypted using the symmetric key 127b and the symmetric ciphering algorithm. Again, key 127b can comprise a master key such as the PSK contemplated by TLS, DTLS, or WiFi standards, and the encryption key for data transmitted can be derived from the master key.

Although not depicted in Figure 4a, a message 412 can also include a set of cryptographic parameters 191e associated with PKe 409b sent in message 412. Symmetric ciphering key 223a can be derived from key 223 using a key derivation function 443 (in Figure 4b below). The ciphertext can be an encrypted form of application data 112a. The use of “{ }” symbols in Figure 4a is to illustrate that the data transmitted is encrypted with the key 223a and the key 223a is not normally transmitted in a message 412. For embodiments where a recorded symmetric key 127b is used as the ciphering key 223a (and RU 109b does not derive an

ephemeral PKI key pair including key 409a), then key PKe 409a can be omitted from a message 412.

Server A 108 can receive message 412 through wireless network 102 and IP network 111. Server A 108 can conduct a series of steps to process the data in message 412. Server A 108 can conduct a step 413 to read SRU identity 109i and select SRU public key SRU PK 109w which corresponds to SRU private key SRU SK 109p recorded and used by RU 109b. In step 413, server A 108 can also use SRU identity 109i to select a set of cryptographic parameters 191e associated with SRU public key SRU PK 109w, and SRU public key SRU PK 109w could also be recorded in a certificate with the cryptographic parameters 191e. At step 414, server A 108 can conduct an ECDH key exchange 442 using (i) the received ephemeral public key PKe 409a from message 412 and (ii) the SRU-SP private key SRU-SP SK 106a in order to mutually derive the same shared secret key 223 with RU 109b. A key exchange step 414 for server A 108 is also depicted below in connection with Figure 4b.

Note that the ECDH key exchange 442 in step 414 could also use as input the SRU public key SRU PK 109w in conjunction with the received ephemeral ECC public key PKe 409b (in other words, an elliptic curve point addition operation on key 409b and key 109w). In this manner, successfully mutually deriving key 223 can require (A) that server A 108 holds or records all of keys 409b, 109w and 106a, and (B) that RU 109b and SRU 109 hold or records all of the corresponding keys for the private key of 409a, 109p, and the public key of 106b. Thus, and ECDH key exchange 442 for steps 414 and steps 409 can support mutual authentication, since only entities on each side with the respective keys listed in the sentence above can mutually derive shared secret key 223 (e.g. SRU 109 must record private key 109p and server A 108 must record private key 106a). In other words, from the perspective of RU 109b, key 223 can be derived from $\{[(\text{private keys } 409a + 109p) \bmod n] * 106b\}$. As one exemplary embodiment, when using the named elliptic curve secp128r1, the value of n can comprise the decimal number:

340282366762482138443322565580356624661

From the perspective of server A 108, key 223 can be derived from $[(409b + 109w) * 106a]$, where the “+” sign indicates an elliptic curve point addition operation with the two public keys. Note that in exemplary embodiments, mutually derived key 223 is not used directly with a symmetric ciphering algorithm, but rather nodes input key 223 into a key derivation function 443 in Figure 4b below (which could use a secure hash algorithm) in order to calculate a symmetric ciphering key 223a. The use of an ephemeral PKI key pair for step 409 and 414 can provide forward secrecy, such that new secret shared keys 223 are periodically derived and the

encryption is not limited to depending on PKI keys for SRU 109 and SPU-SP 106. In some exemplary embodiments, the use of an ECDH key exchange 442 in step 409 can be omitted, and RU 109b and server A 108 could use recorded symmetric key 127b depicted and described as a symmetric ciphering key instead of using a derived key 223.

5 At step 415, server A 108 can use the symmetric ciphering algorithm from a set of cryptographic parameters recorded with SRU ID 109i in order to decrypt the ciphertext in message 412 using the mutually derived key 223 to calculate a symmetric ciphering key 223a from a key derivation function 443. Server A 108 can then conduct a decompression step 416 using the same algorithm as RU 109b (except of decompression, such as using “gzip -d”) in
10 order to read the plaintext values of device identity 101a and http_request 191b. After a successful step 416, although not depicted in Figure 4a, server A 108 could send RU 109b a confirmation of the successful receipt and/or processing of data from RU 109b in message 412. With confirmation of successful transfer of message 412 with http_request 191b (in encrypted and compressed form), RU 109b can then perform step 408b to delete the file 191y for
15 http_request 191b from directory 191x since the file has been transferred to server A 108. RU 109b could also then append a log_file 306 regarding the successful transmission of http-request 191b.

Server A 108 can then establish a TLS session 114 with server B 110a. TLS session 114 can comprise a regular TLS session such as TLS version 1.3 with mutual authentication from
20 certificates 110c and 106c. Note that the use of a TLS session between server B 110a and server A 108 instead of between server B 110a and RU 109b solves the significant problem of bandwidth overhead in order to conduct a TLS session between RU 109b and server 110a. Likewise, a DTLS session would have similar overhead and both DTLS and TLS may not be suited for long sleep periods of device 101, where application data 112a could be received in
25 exemplary timeframes of several times a day to every few days. In other words, the requirements for DTLS and TLS may mean that resumption of previous sessions may not be available from RU 109b using DTLS or TLS with relatively long sleep times and the sessions could regularly expire and require re-establishment for each set of application data 112a such as http_request 191b, with resulting higher overhead and bandwidth utilization.

30 In other words, the bandwidth between RU 109b and server A 108 could be as low as a few kilobits per second for the uplink (such as if RU 109b and device 101 are in a region or location in wireless network 102 with poor RF connectivity), and thus a TLS or DTLS session for each http_request 191b between RU 109b and server B 110a would not be feasible (or could require excessive power drain on a battery 101n). As depicted in Figure 4a, the collection of

steps and messages beginning with a step 408a through step 114 can comprise a step 314. A step 314 was also depicted and described in connection with Figure 3c above.

After setup of TLS session 114, server A 108 could send server B 110a a message 418, where message 418 could be (i) encrypted by the TLS session and (ii) include device identity 101i and http_request 191b. Device identity 101i could be included in the http_request 191b. Although TLS is depicted for session 114 in Figure 4, other secure networking technologies could be used as well, such as a VPN, SSH, or IPsec. Or, a TLS session 114 could be omitted if data 191a is encrypted internally by device 101 with a mutually shared secret key held by server B 110a and device 101 outside of SRU 109.

Server B 110a could receive message 418 and process the message. In step 419a, server B 110a could remove encryption by TLS session 114 and read the plaintext values for device identity 101i and http_request 191b. At step 419b, server B 110a could process the plaintext data from step 419a and generate a http_response 192b. Server B 110a could then send server A 108 a message 420, where message 420 can include http_response 192b (as encrypted data in the TLS session 114) and a server B digital signature 307. Although not depicted in Figure 4a, message 420 could also include device identity 101a so that server A 108 could keep track of which http_response 192b belongs with which SRU 109 (and server A 108 could record a table mapping SRU identities 109i to device identities 101a).

In addition, although the response from server B 110a is depicted as a http_response 192b, the response could comprise text or binary data from server B 110a in response to receiving and processing the data in http_request 191b (which could also be text or binary data). A server B digital signature 307 could also be omitted from a message 420 for embodiments where RU 109b and device 101 trusts server A 108. Or, server B digital signature 307 could be included within http_response 192b. Note that the use of a server B digital signature 307 can be helpful for device 101 to authenticate the data in http_response 192b for exemplary embodiments where the transfer of application data 112a from device 101 to wireless network 102 does not use TLS or does not establish “end to end” encryption and authentication between device 101 and server B 110a.

At step 421, server A 108 could receive message 420 and process the message. At step 421, server A 108 could remove encryption through the TLS session 114 with server B 110a. Server A 108 could read the plaintext data comprising http_response 192b and server B digital signature 307. Note that the inclusion of server B digital signature 307 in a TLS session 114 would not normally be required with TLS sessions 114 with conventional technology, since the server originating the message (e.g. server B 110a) would already be authenticated. Thus, the

inclusion of server B digital signature 307 within TLS session 114 can be different than conventional technology. As mentioned in the paragraph above, message 420 can also include device identity 101a for the http_response 192b. In exemplary embodiments, http_request 191b can include a random number generated by device 101 and server B digital signature 307 can be over both the data within http_response 192b and the random number generated by device 101 and received in http_request 191b.

At step 422, server A 108 can compress the decrypted, plaintext http_response 192b using standard compression libraries such as zip or gzip, and other possibilities exist as well. Server A 108 can also select or identify an RU 109b for http_response 192b using device identity 101a (where server A 108 can include a table mapping device identity 101a to SRU identity 109i). At step 423, server A 108 can encrypt http_response 192b using the mutually derived shared secret key 223 and a resulting symmetric ciphering key 223a. The encryption algorithm used with an encryption step 423 can be specified in a set of cryptographic parameters, such as specifying the use of AES encryption with 128 bit symmetric ciphering keys, although other possibilities exist as well. The output of symmetric ciphering at a step 423 can comprise a ciphertext.

Server A 108 can then send RU 109b a message 424 via wireless network 102, where message 424 can include the ciphertext from a step 423. Message 424 can include encrypted and compressed data for http_response 192b received from server B 110a, where the ciphertext is encrypted using mutually derived shared secret key 223. As mentioned above, the symmetric ciphering key 223a can comprise a number derived from key 223 using a key derivation function from a set of cryptographic algorithms equivalent to cryptographic algorithms 109q recorded by SRU 109. Message 424 can also include server B digital signature 307. The http_response 192b in a message 424 can comprise server B response 112b data. Note that server B digital signature 307 can be included in the ciphertext for a message 424, which again is different from conventional technology where a server digital signature is not normally included in a ciphertext. Or, server B digital signature could be sent in a message 424 outside or external to the ciphertext (e.g. not encrypted with key 223a).

At step 425, RU 109b can receive the message 424 and process data within the message. RU 109b can conduct a step 425, which can comprise server A 108 using the mutually derived shared secret key 223 and symmetric ciphering key 223a to decrypt data within message 424. RU 109b can then decompress the data in a step 426 using standard libraries such as zip or gzip in order to read decompressed, decrypted plaintext data from message 424 from server A 108. At step 427, RU 109b and SRU 109 can then write files 192y to directory 192x in shared