



DECLARATION OF GORDON MACPHERSON

I, Gordon MacPherson, am over twenty-one (21) years of age. I have never been convicted of a felony, and I am fully competent to make this declaration. I declare the following to be true to the best of my knowledge, information, and belief:

1. I am the Director, Board Governance & Policy Development at The Institute of Electrical and Electronics Engineers, Incorporated ("IEEE").
2. IEEE is a neutral third party in this dispute.
3. I am not being compensated for this declaration and IEEE is only being reimbursed for the cost of the article I am certifying.
4. Among my responsibilities as Director of Board Governance & Policy Development, I act as a custodian of certain records for IEEE.
5. I make this declaration based on my personal knowledge and information contained in the business records of IEEE.
6. As part of its ordinary course of business, IEEE publishes and makes available technical articles and standards. These publications are made available for public download through the IEEE digital library, IEEE Xplore.
7. It is the regular practice of IEEE to publish articles and other writings including article abstracts and make them available to the public through IEEE Xplore. IEEE maintains copies of publications in the ordinary course of its regularly conducted activities.
8. The articles below have been attached as Exhibits A-B to this declaration:

Exhibit A	J. Park, K. Baek and C. Kang, "Secure Profile Provisioning Architecture for Embedded UICC," 2013 International Conference on Availability, Reliability and Security, Regensburg, Germany, 2013, pp. 297-303, doi: 10.1109/ARES.2013.40.
Exhibit B	P. E. Abi-Char, A. Mhamed and B. El-Hassan, "A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications," The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007), Cardiff, UK, 2007, pp. 235-240, doi: 10.1109/NGMAST.2007.4343427.

9. I obtained a copy of Exhibit A through IEEE Xplore, where it is maintained in the ordinary course of IEEE's business. Exhibit A is a true and correct copy of the Exhibit, as it existed on or about November 12, 2025.
10. I obtained a copy of Exhibit B through IEEE Xplore, where it is maintained in the ordinary course of IEEE's business. Exhibit B is a true and correct copy of the Exhibit, as it existed on or about November 12, 2025.
11. The article and abstract from IEEE Xplore show the dates of publication. IEEE Xplore populates this information using the metadata associated with the publication.
12. J. Park, K. Baek and C. Kang, "Secure Profile Provisioning Architecture for Embedded UICC," was published in 2013 International Conference on Availability, Reliability and Security, held on September 2-6, 2013, in Regensburg, Germany. Copies of the conference proceedings were made available no later than the last day of conference. This article was added to the IEEE digital library Xplore on November 7, 2013, as shown from the abstract from IEEE Xplore, and was publicly available for download from IEEE Xplore by November 7, 2013, and is currently available for public download from the IEEE digital library, IEEE Xplore.
13. P. E. Abi-Char, A. Mhamed and B. El-Hassan, "A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications," was published in The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007), held on September 12-14, 2007, in Cardiff, United Kingdom. Copies of the conference proceedings were made available no later than the last day of conference. This article was added to the IEEE digital library Xplore on October 8, 2007, as shown from the abstract from IEEE Xplore, and was publicly available for download from IEEE Xplore by October 8, 2007, and is currently available for public download from the IEEE digital library, IEEE Xplore.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

I declare under penalty of perjury that the foregoing statements are true and correct.

Date: 11/13/2025

Signed by:
Gordon Macpherson
845214E4A5C0470...

EXHIBIT A

Secure Profile Provisioning Architecture for Embedded UICC

Jaemin Park, Kiyoung Baek, Cheoloh Kang
System R&D Division
The Attached Institute of ETRI
Daejeon, Republic of Korea
Email: {jmpark, cloud, cyberkan}@ensec.re.kr

Abstract—Embedded UICC (eUICC) is a new form of UICC, soldered into a device during manufacturing. On the contrary to the traditional UICC, the eUICC is not fully controlled by one specific MNO (Mobile Network Operator) since not removable physically from the device and not issued by the MNO. Thus, the profiles necessary for its operations should be provisioned remotely into the eUICC by new entity. For this remote provisioning, SM (Subscription Manager) is newly introduced by the standardization organization. However, this new ecosystem around eUICCs can cause tremendous security issues unless thorough consideration of security is accompanied during the standardization because the profiles usually include the security-sensitive information.

In this paper, a novel secure profile provisioning architecture for eUICCs is proposed. Our architecture mainly defines the overall architecture of the secure profile provisioning for eUICCs.

Keywords—Security, Provisioning, Embedded UICC, Subscription Manager, Secure Architecture

I. INTRODUCTION

Embedded UICC (hereafter eUICC) [2] is a new form of UICC (Universal Integrated Circuit Card) [1], soldered into a device during manufacturing. The eUICC was initially considered to be utilized as the same roles of UICC for the M2M (Machine-to-Machine) device so as to endure the hostile environments such as high temperature, humidity, etc. or be adopted into the small device such as smart meter, etc. These days, the fields of its usages are being considered to be extended to the CEDs (Consumer Electronic Devices) for the smaller form factor to save the physical space of a device.

The traditional UICC is pre-provisioned by the SIM Vendor on the behalf of MNO by using the the MNO's data and applications (hereafter profiles) such as NAA(Network Access Application) [1] with related data (e.g. IMSI (International Mobile Subscriber Identity), keys, authentication algorithm, etc.) and fundamental VASs (Value-Added Services) [4]. That is, the UICC is tightly coupled with and fully controllable by one specific MNO. On the contrary to this, the eUICC is tightly coupled with its installed device rather than MNOs and not fully controlled by one specific MNO since not removable physically from the device. Thus any MNO or even other parties wish to and should be able to provision their profiles remotely into the eUICC. This remote profile provisioning is also essential for the eUICC to support the handset switching, national swapping and international swapping. The inherent feature of the eUICC and the absence of the control entity

for the eUICC inevitably leads to a new ecosystem, and the SM (Subscription Manager) is newly defined to manages and controls the profiles by the standardization [2], [3]. According to [2], SM is further divided into two roles, SM-DP (Data Preparation) for the profile generation and SM-SR (Secure Routing) for the profile transportation, and each role can be performed by separate physical or logical entities. Because the profile is security-sensitive to the SM-DP itself and the customer, it is obvious that each SM-DP is hardly willing to reveal their generated profile to others even SM-SR.

In this new ecosystem around eUICC, the tremendous security issues can be raised unless thorough consideration of security about the profile provisioning is accompanied during the standardization and the technical development. Since the profile includes the security-sensitive information, without protection during provisioning, the sensitive information can be revealed to malicious adversaries or competitors of business. Furthermore, a newly introduced SM essentially raises the security threats because of its right to control eUICCs and roles to provision the profiles. The profile provisioning should also consider the inherent non-removable feature and the diversity of capability of eUICC, otherwise there should be a repository to manage the huge amount of information related to the profile provisioning.

Motivated by above security concerns, we propose a novel secure provisioning architecture to guarantee the security of the profile provisioning regardless of the capabilities of eUICCs. The rest of this paper is organized as follows: In Section II, we present the preliminaries about the eUICC provisioning ecosystem, the standardization status and the related works. In Section III, the problem statement is described. In Section IV, we propose our profile provisioning architecture and internal architecture of eUICC. In Section V, the analysis of our proposed architecture is given. Finally, the conclusions and further works are given in Section VI.

II. PRELIMINARIES

A. eUICC Provisioning Ecosystem

The eUICC provisioning ecosystem mainly consists of eUICCs, eUICC Vendors, Devices and SMs (SM-SRs and SM-DPs) [2], [4] as shown in Fig. 1. SM is divided into two parties; one is SM-SR for secure routing and the other is SM-DP, the owner or generator of security-sensitive information, for data preparation. As shown in Fig. 1, small numbers of SM-SRs are interworking with each other and SM-DPs delegate the roles

of secure routing of the sensitive information to their trusted SM-SRs. For the provisioning and management, one eUICC usually interworks with one SM-SR. Because of their specific roles, one SM-SR can interwork with several SM-DPs for secure routing of profiles generated by SM-DPs to eUICCs. For scalability, several SM-SRs can interwork each other to extend the manageability to other eUICCs associated with other SM-SRs. For secure routing, SM-SRs usually utilize the SM-SR Credentials, installed in eUICCs during the manufacturing and shared by eUICC Vendors.

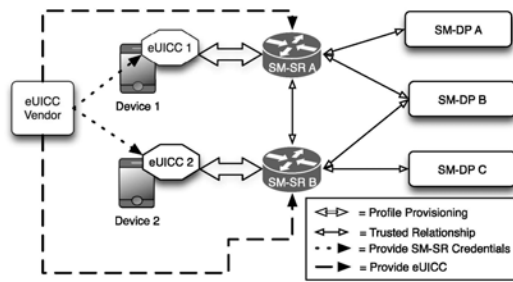


Fig. 1. eUICC Profile Ecosystem

B. Standardization and Related Works

The eUICC is standardized by the ETSI (European Telecommunications Standards Institute) SCP (Smart Card Platform), and the working document is ETSI TS 103.383 V12.0.0 [2]. This standard defines the use cases and requirements about the eUICC and interworking with SMs. GSMA (Groupe Spéciale Mobile Association) publishes the document about requirements for eUICC, which defines a number of requirements and use cases for developing eUICC systems [3]. These standards only define the conceptual models, use cases and requirements. Since only conceptual architecture of eUICC is available, there exist few works about eUICC. According to [4], several patents and proofs of concept have been released, but those are not related to the security or published in public to be reviewed or analyzed.

As mentioned above, the current status of standardization is still immature. Thus, any detail and published secure profile provisioning directly related to eUICC cannot be found, but only few works which describes the enhanced business process for a flexible MCIM (M2M Communication Identity Module) [7] and several works related to the provisioning for UICCs or smart cards can be found [6], [8]–[13]. In [7], the business process related to M2M is mainly focused, therefore no detail consideration of security can be found. In [6], the methodology to utilize current UICC technology to provision UICC-based applications after issuing of UICCs is proposed where GP Secure Channel Protocol (hereafter SCP) is applied to the payment applications for post-distribution provisioning and personalization with the utilization of public key-based key agreement between Bank and UICC. In [8], [9], the On-board Credentials (ObCs) are proposed to utilize the general-purpose secure hardware as credentials by provisioning the credentials into ObC devices. In [10]–[13], new paradigm of smart card ownership where the ownership is delegated

to users not card issuers and its related security protocols are proposed. However, these approaches are difficult to be directly applied to the eUICC provisioning ecosystems even though some portions of works can be utilized to design other architectures or detail protocols. The reasons mainly include; no consideration of fundamental eUICC ecosystem, essential key separation among SM-SRs and SM-DPs, various kinds of capabilities which eUICCs possess, and protection of that capability information. Therefore, we consider the way to apply the de-facto standard for profile provisioning, GP SCP [5] to the eUICC provisioning ecosystem for comparison.

C. Method to Apply GP SCP to eUICC

Before describing the methodology, we first need to explain SD (Security Domain) and SCP briefly. SD is an application running inside the UICC to act as the on-card representatives of off-card entities. SD mainly supports security services such as key handling, encryption, decryption, digital signature, etc. and can be divided into two categories; ISD (Issuer SD) is the primary, mandatory on-card representative of the Card Issuer (e.g. MNO) and SSDs (Supplementary SDs) are additional, optional on-card representatives of Application Providers (e.g. Bank). SCP provides the secure communication channel between these SDs and off-card entities to manage UICCs and their applications securely. SCP usually provides the confidentiality, entity authentication, data origin authentication and integrity. By sending and receiving the APDU (Application Protocol Data Unit) commands between SD and off-card entity via SCP, the content can be loaded and installed into the UICC.

To apply the GP SCP for eUICC, we need to define the relationship between SDs and components of eUICC provisioning ecosystem; ISD is the on-card representative of SM-SR and SSD is of SM-DP. To utilize SCP connections, the ISD key between eUICC and SM-SR should be inserted into the eUICC during manufacturing processes or other secure manners. The SSD key should be shared between SM-SR and SM-DP in advance of profile provisioning using additional secure manners or other key agreement protocol between SSD and SM-DP should be utilized mentioned in [6].

Using above circumstances, the method to apply the GP SCP can be depicted as shown in Fig. 2. The eUICC establishes the network access to SM-SR. SM-SR establishes the 1st SCP session with ISD of eUICC using ISD key. SM-SR installs new SSD with one of the privileges like Authorization Management or Delegated Management into eUICC by communicating with ISD of eUICC since only the SSDs of the listed privileges can load and install the profile into the eUICC. After this, SM-SR also inserts the SSD key of SM-DP into the installed SSD for SM-DP if the key agreement protocol mentioned in [6] is not utilized. The 1st SCP session is completed and terminated. Now, the eUICC establishes the additional network access to SM-DP. SM-DP establishes a new SCP session with the SSD using the SSD key. SM-DP sends the blocks of profile and APDU commands to the SSD via the 2nd SCP session. If the SSD has the privilege of Delegated Management, SM-SR should send a Token to SM-DP for the authorization. After successful completion of profile provisioning, the 2nd SCP session would be terminated.

Even though this approach can work properly, applying the GP SCP directly into eUICC provisioning ecosystem

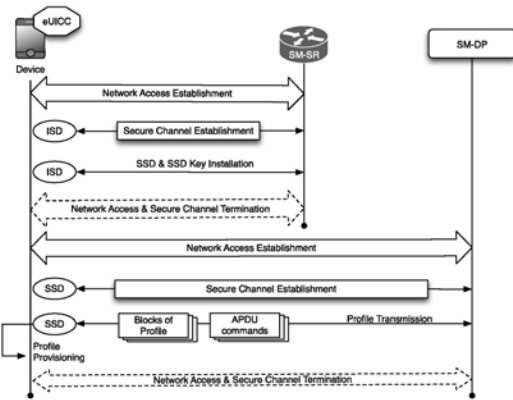


Fig. 2. GP SCP for eUICC Profile Provisioning

has several drawbacks with respect to scalability, efficiency, security and flexibility as follows:

Scalability - To share the SSD key, SM-SR and SM-DP should implement additional secure channel each other. These additional and inconvenient efforts can hinder the scalability.

Efficiency - Two communication channels and two SCP sessions should be guaranteed. That is, SM-DP also needs to implement the communication infrastructure to interwork with eUICCs. Additional communication channel and SCP connection can be the burden of eUICCs and SMs.

Security - Since the initial SSD key is shared between SM-SR and SM-DP, the blocks of profile transferred by SCP can be revealed to other entities if installing new SSD key into eUICC by SM-DP is not performed before profile provisioning.

Flexibility - Only specific cryptographic algorithms defined in GP specification can be utilized for profile provisioning. This limitation can be dangerous if security threats of the algorithms are found. Furthermore, the variation due to the local regulation about the cryptographic algorithms is impossible.

III. PROBLEM STATEMENT

In this section, we define the requirements of the profile provisioning architecture for eUICC.

A. General Requirement

As mentioned above, profile provisioning architecture for eUICC should be proposed with the consideration of new provisioning ecosystem. Furthermore, only minimum processing overhead for the security-related operations should be a burden to the eUICC during the profile provisioning since the eUICC is also the resource-constraint medium. Finally, to address the drawbacks of the existing technology, the scalability, the efficiency, and the flexibility in Section II should be enhanced.

B. Security Requirement

Because the profile mostly includes the security-sensitive information, the protection should be supported. Therefore,

the following security requirements can be guaranteed: The following security requirements should be guaranteed in accordance with the requirements in [5]:

Mutual Authentication - eUICC and SM-DP prove their authenticities each other through cryptographic exchanges.

Confidentiality - profiles being transmitted to eUICC by SM-DP are only available to those authorized entities.

Data Integrity - data being exchanged between eUICC and SM-DP has not been altered by unauthorized entities.

Data Origin Authentication - eUICC and SM-DP ensure that data actually came from the authorized entities, SM-DP and eUICC, respectively.

Non-Exposure of Key - credentials between eUICC and SM-DP are expose only to directly involved entities in the profile provisioning; eUICC and SM-DP.

Verification of eUICC's Information - information of eUICC sent to other entities is able to be verified by the receivers.

IV. PROPOSED PROFILE PROVISIONING ARCHITECTURE

In this section, we propose our architecture, Secure Profile Provisioning Architecture for eUICC (hereafter SPA) to address the issues raised in Section III and meet the security requirements listed below. SPA mainly defines the overall architecture for the secure profile provisioning. Therefore, the changeable security protocols like secure communication protocol and key agreement protocol are out of scope.

A. Terminologies

We define the terminologies in Table I. Some terminologies are come from [2] and reproduced here for simplicity.

TABLE I. TERMINOLOGIES

Terminology	Description
eUICC ID	Identification for eUICC to identify a particular eUICC
(eUICC Certification Center	The trusted third party to evaluate eUICCs (e.g. GSMA's SAS (Security Accreditation Scheme))
Provisioning Profile	The profile to be utilized to access SMs for the profile provisioning via mobile networks
SM-SR Credential	The credential to establish the secure communication channel between eUICC and SM-SR
SM-DP Credential	The credential to protect the profile generated by SM-DP
Capability Information	The information stored in the eUICC, including supporting cryptographic algorithms, key agreement protocols, the capacity information, etc.
Key Agreement Module (KAM)	The software running in the eUICC to perform the key agreement protocol to agree SM-DP Credentials with SM-DP
Device Module (DM)	The software running in the Device to perform the roles of the intermediary between eUICC and SM-SR
Profile Installer (PI)	The software running in the eUICC to perform the decryption and verification of the protected profile blocks from SM-DP and provision the profile blocks
Profile Manager (PM)	The software running in the eUICC to manage profiles as blocks of encrypted data and has relationship with SM-SR

B. Secure Profile Provisioning Architecture (SPA)

SPA mainly consists of two procedures; Pre-Provisioning Procedure and Secure Profile Provisioning Procedure.

1) *Pre-Provisioning Procedure*: SPA involves the pre-provisioning procedure, conducted during the eUICC-based device manufacturing. Through this procedure, SM can collect the necessary information for SPA, and the verification of the capability information of the eUICC can be possible when performing the secure profile provisioning in the field. The detail of this pre-provisioning procedure is depicted in Fig. 3.

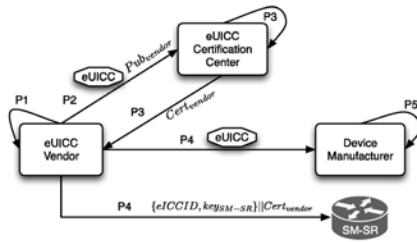


Fig. 3. Pre-Provisioning Procedure

Step P1 - The eUICC Vendor develops eUICCs. During this, the SM-SR Credentials (key_{SM-SR}) and the capability information ($CI_{installed}$) of each eUICC are installed into each eUICC. Then, the eUICC Vendor generates the public key pair, and installs the private key ($Priv_{vendor}$) into the eUICCs.

Step P2 - The eUICC Vendor requests the evaluation of the eUICCs to the eUICC Certification Center with the public key (Pub_{vendor}) of Step P1.

Step P3 - Only after the successful evaluation, the eUICC Certification Center responds the results with the certificate ($Cert_{vendor}$) by using Pub_{vendor} .

Step P4 - The eUICC Vendor can install the received certificate into the eUICCs, and sends the developed and evaluated eUICCs to the device manufacturers. Simultaneously, the eUICC Vendor sends the information to SM-SR required for SPA such as the certificate, the eUICC IDs, the SM-SR Credentials, the mapping table between eUICC IDs and SM-SR Credential indexes, etc. in the secure manner.

Step P5 - The Device Manufacturer develops the eUICC-based devices using the received eUICCs.

2) *Secure Profile Provisioning Procedure*: After the pre-provisioning procedure, the eUICC-installed devices are deployed in the field. For their operations, the profiles should be provisioned to the eUICCs. Fig. 4 depicts the secure profile provisioning procedure in detail. For the simplicity, we only denote one eUICC and one SM (one SM-SR and one SM-DP); however, the number of the entities can be extended depending on the needs of the market.

Step 1 - The eUICC initially accesses SM-SR using the Provisioning Profile. In addition, the eUICC can access SM-SR via WiFi, USB, etc. without the Provisioning Profile. Also, the provisioned profile for accessing the mobile network can be utilized to reach to SM-SR.

Step 2 - After the establishment of the communication channel, the eUICC sends the message including the eUICC ID ($eUICCID$) and its capability information ($CI_{proposal}$) to SM-SR for the capability negotiation. The eUICC ID and capability

information are signed by the eUICC Vendors private key, and this signed value needs to be added to the message. If there is no KAM installed onto the eUICC, the eUICC can replace the key agreement protocol of the CI message with proper value such as null, blank, error message, etc. to inform SM that the installation is required. Simultaneously, the eUICC can also send the eUICC Vendors certificate if necessary.

Step 3 - SM acquires the eUICC Vendors certificate from the eUICC Certification Center or the message sent by the eUICC in Step 2. With that certificate, SM-DP verifies the signature of the eUICC ID and the capability information.

Step 4 - Then, SM-DP sends the message including the selected capabilities ($CI_{selected}$, the cryptographic algorithms, the key agreement protocol, the secure communication protocol and so forth) and its hashed value to the eUICC via SM-SR if the decision is completed based on the received capability information. The capability information about the secure communication protocol can be chosen through the interworking between SM-SR and SM-DP.

Step 5 - PM of eUICC and the SM-SR establish the secure communication channel based on the selected protocol. The SM-SR Credential can be same as or derived from the master secret installed in Step P1.

Step 5 - The blocks of KAM from SM-DP and the acknowledgement from the eUICC can be sent through the secure communication channel established in Step 5 and installed to the eUICC if the message of Step 2 indicates that the installation is necessary. This can be happened when the change of security policy is required, the vulnerabilities of the cryptographic algorithm used in that module are found, or other reasons are happened.

Step 6 - KAM of eUICC and SM-DP perform the key agreement protocol dynamically to generate the SM-DP Credentials (key_{SM-DP}) through the established secure communication channel between the eUICC and SM-SR.

Step 7 - Using the agreed SM-DP Credentials and the selected cryptographic algorithms, SM-DP encrypts each block of profile ($Profile_i$) and generates its MACed value. Then, SM-DP sends the protected profile blocks to the eUICC via the secure communication channel of SM-SR. PM of eUICC performs the necessary cryptographic operations for the secure communication channel, and then PI of eUICC decrypts and verifies the received profile blocks. If no problem found, the profile blocks are provisioned inside the eUICC by PI. Then, PI sends the acknowledge for each block of profile with its MACed value to SM-DP. This step continues recursively until the provisioning is completed.

C. Proposed Internal Architecture of eUICC

The change of the internal architecture of the eUICC compared to the conceptual architecture in [2] is inevitable to support SPA as depicted in Fig. 5. Since new module, KAM is added to the internal architecture of eUICC, interworking among modules should be considered to reflect this change. DM of the device should be implemented to handle the messages from SM-SR. Handled messages are sent to PI for provisioning, and PI interworks with KAM to retrieve the agreed SM-DP Credentials for decrypting and verifying the

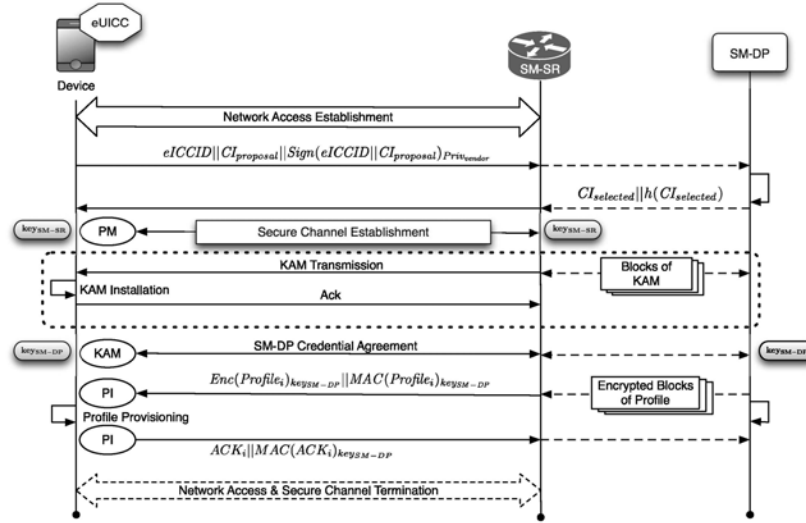


Fig. 4. Secure Profile Provisioning Procedure

protected profile blocks. After completing of cryptographic operations, PI provisions the profile into the eUICC.

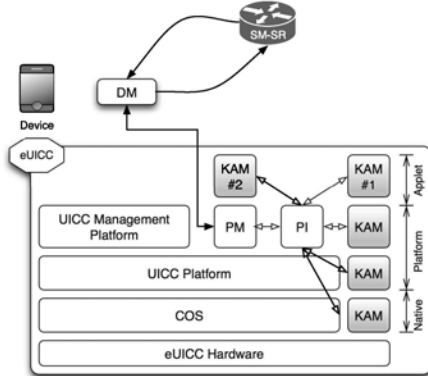


Fig. 5. Internal Architecture of eUICC

1) **Key Agreement Module (KAM):** KAM is software running inside the eUICC to perform the key agreement protocol, which is already known by eUICC and SM-DP after the capability negotiation of SPA, with SM-DP. KAM is mainly utilized to agree the SM-DP Credentials on demand. For the security, the SM-DP Credentials should not be revealed to any party, even SM-SR, except for eUICC. To accomplish this, KAM is designed and applied to SPA. KAM can provide the asymmetric algorithm-based key agreement protocol, or possess the master secrets of SM-DP Credentials, which are accessible only after the authentication and where SM-DP Credentials are derived afterward.

KAM can be different from each SM-DP if the capabilities of the eUICC are affordable. That is, depending on the security policies, the security threats found in some specific algorithms, and other reasons, different KAMs can be installed. Depending on the selected cryptographic algorithms for protecting profiles, KAM should be able to agree all kinds of SM-DP Credentials. KAM can be various forms like installable applet, native program of COS (Chip OS) or component of the UICC platform (e.g. Javacard Platform or GP) as shown in Fig. 5.

2) **Capability Information:** Depending on the hardware specification and the implementations, the eUICC can be different from each other with respect to supporting cryptographic algorithms, affordable key agreement protocols, affordable secure communication protocols, total and available memory capacities, existences of KAM or related information, etc. If SM should manage all of this information, necessary storage can be drastic. Then, this complexity yields the increment of error and management cost. Therefore, this capability information should be stored inside the eUICC and be retrieved by the external entities if necessary. Without the guarantee of the correctness and the verification about this information, SM has no method to convince that the capability of the eUICC is correctly implemented. Therefore, the technical method to verify the capability information sent from the eUICC should be existed. To address these issues, we propose a methodology to store the capability information of each eUICC inside itself and verify this information if necessary.

V. ANALYSIS

In this section, we evaluate our SPA against the requirements defined in Section III. Furthermore, we compare our SPA with the approach mentioned in Section II.

A. Security Features

SPA provides the secure profile provisioning together with supporting the non-exposure of key and the verification of eUICC's information (capability information).

Mutual Authentication - Depending on the key agreement protocol supported by KAM, the mutual authentication between eUICC and SM-DP can be provided or not. However, there might exist no key agreement protocol without mutual authentication.

Confidentiality - Confidentiality is guaranteed since the profile being transmitted is encrypted by the SM-DP credential.

Data Integrity - Integrity is guaranteed by performing MAC operation for all blocks of profile by SM-DP and all acknowledgements by the eUICC. Therefore, eUICC and SM-DP ensure that data being received has not been altered.

Data Origin Authentication - Data origin authentication is guaranteed by performing MAC operation using the agreed SM-DP Credential for all blocks of profile by SM-DP and all acknowledgements by the eUICC. Therefore, eUICC and SM-DP ensure that all data actually came from SM-DP and eUICC, respectively.

Non-Exposure of Key - Because KAM embeds the shared secret onto the eUICC or performs the key agreement protocol with SM-DP, any SM-DP Credential cannot be shared with other entities.

Verification of Capability Information - Since the capability information of eUICC is signed by eUICC Vendors private key stored inside the eUICC, SMs can ensure that the received capability information was evaluated successfully by eUICC Certification Center and has not been altered.

B. Cryptographic Performance of eUICC

To support SPA, the overhead of the eUICC can be increased since the eUICC should perform additional cryptographic operations. The cryptographic performance can be evaluated by measuring the time required for the eUICC to perform necessary cryptographic operations. Therefore, the overall cryptographic performance is calculated as follows:

$$P(A) = \sum t(f) \quad (1)$$

where $P(A)$ is the overall time for the eUICC to perform A and $t(f)$ is the measured time for the eUICC to perform specific cryptographic operation, f .

Using (1), $P(SPA)$ can be calculated as follows:

$$P(SPA) = P(SCP) + P(KAP) + N\{t(Dec) + t(MAC)\} + t(Sign) + t(h) \quad (2)$$

where N is number of blocks of profile, SCP is Secure Communication Protocol and KAP is Key Agreement Protocol.

If the cryptography algorithm for profile decryption is the symmetric and identical to GP SCP, $P(SCP)$ can be approximated to (3) since secure communication protocols mostly include key agreement and protection of blocks of data with verification of MAC values.

$$P(SCP) \approx P(KAP) + N\{t(Dec_{sym}) + t(MAC)\} \quad (3)$$

Then, (2) can be simplified as follows:

$$P(SPA) \approx 2P(SCP) + t(Sign) + t(h) \quad (4)$$

For better understanding, we compare it with the performance of GP SCP (t) in Section II. The cryptographic performance of GP SCP to be performed by eUICC can be as follows:

$$P(GP) = t = 2P(SCP) \quad (5)$$

By subtracting (5) from (4), we can evaluate the performance increment due to SPA in the case when the symmetric algorithm is utilized for the profile protection. Then, we can conclude that the cryptographic overhead increased by SPA can be as follows:

$$\sigma_{sym} \approx t(Sign) + t(h) \quad (6)$$

where σ_{sym} is the performance difference of the eUICC when the symmetric algorithm is applied.

Suppose that the asymmetric algorithm is utilized for protecting the blocks of profile. Even though not suitable for the large block of profile, the performance review against every possible candidates can be meaningful for new ecosystem. To estimate the overhead of the asymmetric case, we subtract (5) from (2) directly and assign (3), then we acquire following (7):

$$\sigma_{asym} \approx M\{t(Dec_{asym} + t(MAC))\} - N\{t(Dec_{sym} + t(MAC))\} + t(Sign) + t(h) \quad (7)$$

where σ_{asym} is the performance difference of the eUICC when the asymmetric algorithm is applied, M is the number of blocks of profile (length= L) for the asymmetric algorithm (block length= $b1$) and N is the number of blocks of same profile for the symmetric algorithm (block length= $b2$).

Since the block lengths of the symmetric and the asymmetric are different each other, and ciphering and MACing are processed for each block of profile, (7) can be expressed as follows using (6):

$$\sigma_{asym} \approx \sigma_{sym} + \left\lceil \frac{L}{b1} \right\rceil \cdot t(Dec_{asym}) - \left\lceil \frac{L}{b2} \right\rceil \cdot t(Dec_{sym}) + \left\{ \left\lceil \frac{L}{b1} \right\rceil - \left\lceil \frac{L}{b2} \right\rceil \right\} \cdot t(MAC) \quad (8)$$

where $\lceil x \rceil$ is the ceiling function of x .

For the reference, according to the cryptographic performance of the CCP (Crypto Co-Processor) used for (e)UICC [14], we calculate the approximate value of each σ against the profile length, L , as shown in Fig. 6. We consider two cases to estimate σ_{asym} depending on the type of MAC algorithm; CMAC (Cipher-based MAC) or HMAC (Hash-based MAC). As the profile length, L , is increased, all σ_{asym} values are increased linearly. Therefore, for the better performance, the symmetric algorithm for encrypting and decrypting the profile is preferable since the constant overhead (σ_{sym}) is promised.

C. Comparison

We compare SPA with the approach of the GP SCP with respect to the security, the performance and the drawbacks mentioned in II. Besides the improved security, SPA overcomes

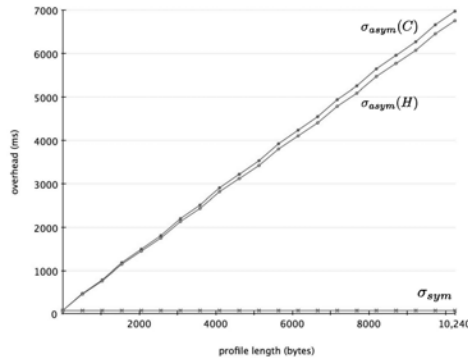


Fig. 6. Cryptography Performance of eUICC

all drawbacks of GP SCP with reasonable overhead, σ of the performance as shown in Table II.

SPA does not necessitate the pre-shared secret between SM-SR and SM-DP since SM-DP Credential is generated on demand. As this inconvenient procedure is removed, the scalability is improved. SM-DPs do not have to implement the communication facilities to be accessed by eUICCs directly since one communication channel and one SCP session with SM-SR are required. This can reduce the management cost of eUICC systems and the network overhead of eUICCs. By utilizing KAM and the capability information of eUICC, SPA supports various kinds of security protocols if eUICC is affordable. Therefore, diverse key agreement protocols, secure communication protocols and cryptographic algorithms can be applied depending on the circumstances around the eUICC.

TABLE II. COMPARISON

Criteria		GP SCP	SPA
Security	Mutual Authentication	O	O
	Confidentiality	O	O
	Data Integrity	O	O
	Data Origin Authentication	O	O
	Non-Exposure of Key	Δ	O
	Verification of Capability Information	$\times$	O
Performance	Communication Channel	2	1
	Secure Communication Channel	2	1
	Cryptographic Overhead of (e)UICC	t	$t + \sigma$
General	Scalability	Δ	O
	Efficiency	Δ	O
	Flexibility	Δ	O

VI. CONCLUSION AND FURTHER WORKS

In this paper, we propose a novel secure profile provisioning architecture, SPA for the eUICC. For the profile provisioning, we primarily defines the overall architecture of the secure profile provisioning for eUICC. To protect the profile and overcome the drawbacks of current UICC technology, we utilize the capability negotiation using the capability information of eUICC and the special module, KAM, to agree SM-DP Credentials dynamically during the profile provisioning procedures. To show the excellence of SPA, we analyze our

SPA against security, cryptographic performance of eUICC, scalability, efficiency and flexibility. Our analysis depicts that SPA is superior to the methodology to apply GP SCP with reasonable overheads of eUICC if the symmetric algorithm for protecting profile is applied.

As further works, we intend to study about new secure communication protocol and key agreement protocol optimized for eUICC systems. This study should mainly deal with new circumstances raised by the eUICC provisioning ecosystem such as the inevitable remote management of numerous eUICCs, the security concerns for all related entities around, etc.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

REFERENCES

- [1] ETSI TR 102 216 V3.0.0, *Smart cards; Vocabulary for Smart Card Platform specifications*, ETSI, 2003.
- [2] ETSI TS 103 383 V12.0.0, *Smart Cards; Embedded UICC; Requirements Specification*, ETSI, 2013.
- [3] GSMA V1.0, *Embedded SIM Task Force: Requirements & Use Cases*, GSMA, 2011.
- [4] CSMG, *Reprogrammable SIMs: Technology, Evolution and Implications*, tech. report, CSMG, TMNG Global, 2012.
- [5] *GlobalPlatform, Card Specification Version 2.2.1*, GlobalPlatform, Jan. 2011.
- [6] V. Alimi and M. Pasquet, "Post-Distribution Provisioning and Personalization of a Payment Applications on a UICC-based Secure Element," *Int'l Conf. Availability, Reliability and Security, 2009 (ARES'09)*, IEEE, 2009, pp.701-705.
- [7] H. Bender and G. Lehmann, "Evolution of SIM provisioning towards a flexible MCIM provisioning in M2M vertical industries," *Int'l Conf. Intelligence in Next Generation Networks, 2012 (ICIN2012)*, IEEE, 2012, pp.57-64.
- [8] J.-E. Ekberg, N. Asokan, K. Kostianen and A. Rantala, *On-board Credentials with Open Provisioning*, tech. report, NRC-TR-2008-007, Research Center, NOKIA, 2008.
- [9] N. Asokan and J.-E. Ekberg, "A Platform for OnBoard Credentials," *Financial Cryptography and Data Security, 2008*, Springer-Verlag, 2008, pp.318-320.
- [10] R. N. Akram, K. Markantonakis and K. Mayers, "Application Management Framework in User Centric Smart Card Ownership Model," *Information Security Applications, 2009*, Springer-Verlag, 2009, pp.20-35.
- [11] R. N. Akram, K. Markantonakis and K. Mayers, "A Paradigm Shift in the Smart Card Ownership Model," *Int'l Conf. Computational Science and Its Applications, 2010 (ICCSA'10)*, IEEE, 2010, pp.191-200.
- [12] R. N. Akram, K. Markantonakis and K. Mayers, "Simulator Problem in User Centric Smart Card Ownership Model," *Int'l Conf. Embedded and Ubiquitous Computing, 2010 (EUC'10)*, IEEE, 2010, pp.679-686.
- [13] R. N. Akram, K. Markantonakis and K. Mayers, "A Privacy Preserving Application Acquisition Protocol," *Int'l Conf. Trust, Security and Privacy in Computing and Communications, 2012 (TrustCom'12)*, IEEE, 2012, pp.383-392.
- [14] STMicroelectronics, "Smartcard 32-Bit RISC MCU with 32, 64, 96, 128 Kbytes EEPROM, Javacard HW Execution & Cryptographic Library," ST22L128 datasheet, 2004.
- [15] C. Paar and J. Pelzl, *Understanding Cryptography*, 1st Edition, Springer, Jul. 2010 (2nd Printing edition).
- [16] B.Kaliski, PKCS#1: RSA Encryption Version 1.5, RFC2313, Network Working Group, 1998.

Secure Profile Provisioning Architecture for Embedded UICC

Publisher: IEEE

[Cite This](#) PDFJaemin Park ; Kiyoun Baek ; Cheoloh Kang **All Authors****7**Cites in
Papers**51**Cites in
Patents**659**Full
Text Views**Abstract**

Document Sections

- I. Introduction
- II. Preliminaries
- III. Problem Statement
- IV. Proposed Profile Provisioning Architecture
- V. Analysis

Show Full Outline ▾

Authors

Figures

References

Citations

Keywords

Metrics

More Like This

Abstract:

Embedded UICC (eUICC) is a new form of UICC, soldered into a device during manufacturing. On the contrary to the traditional UICC, the eUICC is not fully controlled by one specific MNO (Mobile Network Operator) since not removable physically from the device and not issued by the MNO. Thus, the profiles necessary for its operations should be provisioned remotely into the eUICC by new entity. For this remote provisioning, SM (Subscription Manager) is newly introduced by the standardization organization. However, this new ecosystem around eUICCs can cause tremendous security issues unless thorough consideration of security is accompanied during the standardization because the profiles usually include the security-sensitive information. In this paper, a novel secure profile provisioning architecture for eUICCs is proposed. Our architecture mainly defines the overall architecture of the secure profile provisioning for eUICCs.

Published in: 2013 International Conference on Availability, Reliability and Security**Date of Conference:** 02-06 September 2013**DOI:** 10.1109/ARES.2013.40**Date Added to IEEE Xplore:** 07 November 2013**Publisher:** IEEE**Electronic ISBN:** 978-0-7695-5008-4**Conference Location:** Regensburg, Germany

SECTION I. Introduction

Embedded UICC (hereafter eUICC) [2] is a new form of UICC (Universal Integrated Circuit Card) [1], soldered into a device during manufacturing. The eUICC was initially considered to be utilized as the same roles of UICC for the M2M (Machine-to-Machine) device so as to endure the hostile environments such as high temperature, humidity, etc. or be adopted into the small device such as smart meter, etc. These days, the fields of its usages are being considered to be extended to the CEDs (Consumer Electronic Devices) for the smaller form factor to save the physical space of a device.

The traditional UICC is pre-provisioned by the SIM Vendor on the behalf of MNO by using the the MNO's data and applications (hereafter profiles) such as NAA(Network Access Application) [1] with related data (e.g. IMSI (International Mobile Subscriber Identity), keys, authentication algorithm, etc.) and fundamental VASs (Value-Added Services) [4]. That is, the UICC is tightly coupled with and fully controllable by one specific MNO. On the contrary to this, the eUICC is tightly coupled with its installed device rather than MNOs and not fully controlled by one specific MNO since not removable physically from the device. Thus any MNO or even other parties wish to and should be able to provision their profiles remotely into the eUICC. This remote profile provisioning is also essential for the eUICC to support the handset switching, national swapping and international swapping. The inherent feature of the eUICC and the absence of the control entity for the eUICC inevitably leads to a new ecosystem, and the SM (Subscription Manager) is newly defined to manages and controls the profiles by the standardization [2], [3]. According to [2], SM is further divided into two roles, SM-DP (Data Preparation) for the profile generation and SM-SR (Secure Routing) for the profile transportation, and each role can be performed by separate physical or logical entities. Because the profile is



security-sensitive to the SM-DP itself and the customer, it is obvious that each SM-DP is hardly willing to reveal their generated profile to others even SM-SR.

In this new ecosystem around eUICC, the tremendous security issues can be raised unless thorough consideration of security about the profile provisioning is accompanied during the standardization and the technical development. Since the profile includes the security-sensitive information, without protection during provisioning, the sensitive information can be revealed to malicious adversaries or competitors of business. Furthermore, a newly introduced SM essentially raises the security threats because of its right to control eUICCs and roles to provision the profiles. The profile provisioning should also consider the inherent non-removable feature and the diversity of capability of eUICC, otherwise there should be a repository to manage the huge amount of information related to the profile provisioning.

Motivated by above security concerns, we propose a novel secure provisioning architecture to guarantee the security of the profile provisioning regardless of the capabilities of eUICCs. The rest of this paper is organized as follows: In [Section II](#), we present the preliminaries about the eUICC provisioning ecosystem, the standardization status and the related works. In [Section III](#), the problem statement is described. In [Section IV](#), we propose our profile provisioning architecture and internal architecture of eUICC. In [Section V](#), the analysis of our proposed architecture is given. Finally, the conclusions and further works are given in [Section VI](#).

SECTION II. Preliminaries

A. eUICC Provisioning Ecosystem

The eUICC provisioning ecosystem mainly consists of eUICCs, eUICC Vendors, Devices and SMs (SM-SRs and SM-DPs) [2], [4] as shown in [Fig. 1](#). SM is divided into two parties; one is SM-SR for secure routing and the other is SM-DP, the owner or generator of security-sensitive information, for data preparation. As shown in [Fig. 1](#), small numbers of SM-SRs are interworking with each other and SM-DPs delegate the roles of secure routing of the sensitive information to their trusted SM-SRs. For the provisioning and management, one eUICC usually interworks with one SM-SR. Because of their specific roles, one SM-SR can interwork with several SM-DPs for secure routing of profiles generated by SM-DPs to eUICCs. For scalability, several SM-SRs can interwork each other to extend the manageability to other eUICCs associated with other SM-SRs. For secure routing, SM-SRs usually utilize the SM-SR Credentials, installed in eUICCs during the manufacturing and shared by eUICC Vendors.

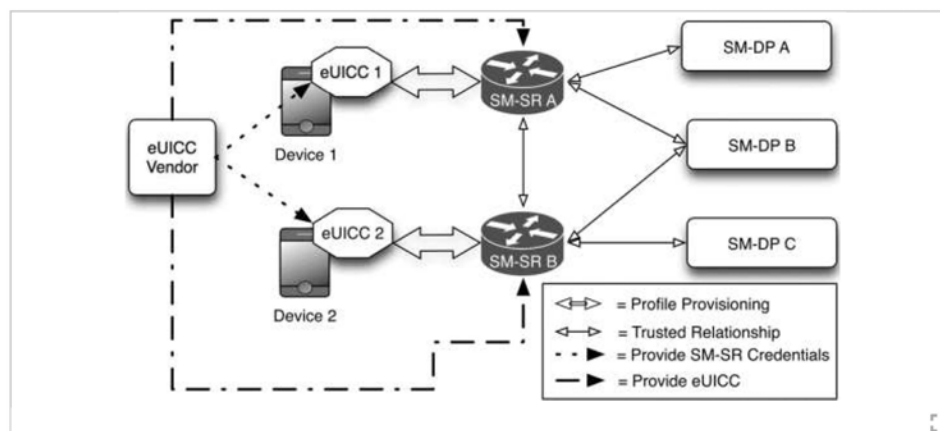


Fig. 1. eUICC Profile Ecosystem

B. Standardization and Related Works

The eUICC is standardized by the ETSI (European Telecommunications Standards Institute) SCP (Smart Card Platform), and the working document is ETSI TS 103.383 V12.0.0 [2]. This standard defines the use cases and requirements about the eUICC and interworking with SMs. GSMA (Groupe Spéciale Mobile Association)

publishes the document about requirements for eUICC, which defines a number of requirements and use cases for developing eUICC systems [3]. These standards only define the conceptual models, use cases and requirements. Since only conceptual architecture of eUICC is available, there exist few works about eUICC. According to [4], several patents and proofs of concept have been released, but those are not related to the security or published in public to be reviewed or analyzed.

As mentioned above, the current status of standardization is still immature. Thus, any detail and published secure profile provisioning directly related to eUICC cannot be found, but only few works which describes the enhanced business process for a flexible MCIM (M2M Communication Identity Module) [7] and several works related to the provisioning for UICCs or smart cards can be found [6], [8]–[13]. In [7], the business process related to M2M is mainly focused, therefore no detail consideration of security can be found. In [6], the methodology to utilize current UICC technology to provision UICC-based applications after issuing of UICCs is proposed where GP Secure Channel Protocol (hereafter SCP) is applied to the payment applications for post-distribution provisioning and personalization with the utilization of public key-based key agreement between Bank and UICC. In [8], [9], the Onboard Credentials (ObCs) are proposed to utilize the general-purpose secure hardware as credentials by provisioning the credentials into ObC devices. In [10]–[13], new paradigm of smart card ownership where the ownership is delegated to users not card issuers and its related security protocols are proposed. However, these approaches are difficult to be directly applied to the eUICC provisioning ecosystems even though some portions of works can be utilized to design other architectures or detail protocols. The reasons mainly include; no consideration of fundamental eUICC ecosystem, essential key separation among SM-SRs and SM-DPs, various kinds of capabilities which eUICCs possess, and protection of that capability information. Therefore, we consider the way to apply the de-facto standard for profile provisioning, GP SCP [5] to the eUICC provisioning ecosystem for comparison.

C. Method to Apply GP SCP to eUICC

Before describing the methodology, we first need to explain SD (Security Domain) and SCP briefly. SD is an application running inside the UICC to act as the on-card representatives of off-card entities. SD mainly supports security services such as key handling, encryption, decryption, digital signature, etc. and can be divided into two categories; ISD (Issuer SD) is the primary, mandatory on-card representative of the Card Issuer (e.g. MNO) and SSDs (Supplementary SDs) are additional, optional on-card representatives of Application Providers (e.g. Bank). SCP provides the secure communication channel between these SDs and off-card entities to manage UICCs and their applications securely. SCP usually provides the confidentiality, entity authentication, data origin authentication and integrity. By sending and receiving the APDU (Application Protocol Data Unit) commands between SD and off-card entity via SCP, the content can be loaded and installed into the UICC.

To apply the GP SCP for eUICC, we need to define the relationship between SDs and components of eUICC provisioning ecosystem; ISD is the on-card representative of SM-SR and SSD is of SM-DP. To utilize SCP connections, the ISD key between eUICC and SM-SR should be inserted into the eUICC during manufacturing processes or other secure manners. The SSD key should be shared between SM-SR and SM-DP in advance of profile provisioning using additional secure manners or other key agreement protocol between SSD and SM-DP should be utilized mentioned in [6].

Using above circumstances, the method to apply the GP SCP can be depicted as shown in Fig. 2. The eUICC establishes the network access to SM-SR. SM-SR establishes the 1st SCP session with ISD of eUICC using ISD key. SM-SR installs new SSD with one of the privileges like Authorization Management or Delegated Management into eUICC by communicating with ISD of eUICC since only the SSDs of the listed privileges can load and install the profile into the eUICC. After this, SM-SR also inserts the SSD key of SM-DP into the installed SSD for SM-DP if the key agreement protocol mentioned in [6] is not utilized. The 1st SCP session is completed and terminated. Now, the eUICC establishes the additional network access to SM-DP. SM-DP establishes a new SCP session with the SSD using the SSD key. SM-DP sends the blocks of profile and APDU commands to the SSD via the 2nd SCP session. If the SSD has the privilege of Delegated Management, SM-SR should send a Token to SM-DP for the authorization. After successful completion of profile provisioning, the 2nd SCP session would be terminated.

Even though this approach can work properly, applying the GP SCP directly into eUICC provisioning ecosystem has several drawbacks with respect to scalability, efficiency, security and flexibility as follows:

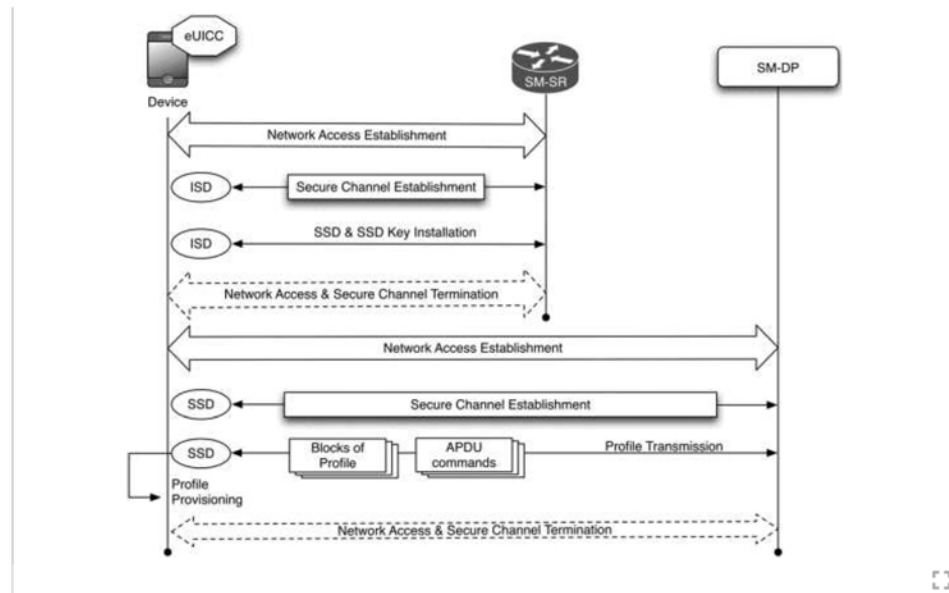


Fig. 2. GP SCP for eUICC Profile Provisioning

Scalability - To share the SSD key, SM-SR and SM-DP should implement additional secure channel each other. These additional and inconvenient efforts can hinder the scalability.

Efficiency - Two communication channels and two SCP sessions should be guaranteed. That is, SM-DP also needs to implement the communication infrastructure to interwork with eUICCs. Additional communication channel and SCP connection can be the burden of eUICCs and SMs.

Security - Since the initial SSD key is shared between SM-SR and SM-DP, the blocks of profile transferred by SCP can be revealed to other entities if installing new SSD key into eUICC by SM-DP is not performed before profile provisioning.

Flexibility - Only specific cryptographic algorithms defined in GP specification can be utilized for profile provisioning. This limitation can be dangerous if security threats of the algorithms are found. Furthermore, the variation due to the local regulation about the cryptographic algorithms is impossible.

SECTION III. Problem Statement

In this section, we define the requirements of the profile provisioning architecture for eUICC.

A. General Requirement

As mentioned above, profile provisioning architecture for eUICC should be proposed with the consideration of new provisioning ecosystem. Furthermore, only minimum processing overhead for the security-related operations should be a burden to the eUICC during the profile provisioning since the eUICC is also the resource-constraint medium. Finally, to address the drawbacks of the existing technology, the scalability, the efficiency, and the flexibility in [Section II](#) should be enhanced.

B. Security Requirement

Because the profile mostly includes the security-sensitive information, the protection should be supported. Therefore, the following security requirements can be guaranteed: The following security requirements should be guaranteed in accordance with the requirements in [5]:

Mutual Authentication - eUICC and SM-DP prove their authenticities each other through cryptographic exchanges.

Data Integrity - data being exchanged between eUICC and SM-DP has not been altered by unauthorized entities.

Confidentiality - profiles being transmitted to eUICC by SM-DP are only available to those authorized entities.

Data Origin Authentication - eUICC and SM-DP ensure that data actually came from the authorized entities, SM-DP and eUICC, respectively.

Non-Exposure of Key - credentials between eUICC and SM-DP are expose only to directly involved entities in the profile provisioning; eUICC and SM-DP.

Verification of eUICC's Information - information of eU-ICC sent to other entities is able to be verified by the receivers.

SECTION IV.

Proposed Profile Provisioning Architecture

In this section, we propose our architecture, Secure Profile Provisioning Architecture for eUICC (hereafter SPA) to address the issues raised in [Section III](#) and meet the security requirements listed below. SPA mainly defines the overall architecture for the secure profile provisioning. Therefore, the changeable security protocols like secure communication protocol and key agreement protocol are out of scope.

A. Terminologies

We define the terminologies in [Table 1](#). Some terminologies are come from [\[2\]](#) and reproduced here for simplicity.

TABLE 1. TERMINOLOGIES

Terminology	Description
eUICC ID	Identification for eUICC to identify a particular eUICC
(eUICC Certification Center	The trusted third party to evaluate eUICCs (e.g. GSMAS SAS (Security Accreditation Scheme))
Provisioning Profile	The profile to be utilized to access SMs for the profile provisioning via mobile networks
SM-SR Credential	The credential to establish the secure communication channel between eUICC and SM-SR
SM-DP Credential	The credential to protect the profile generated by SM-DP
Capability Information	The information stored in the eUICC, including supporting cryptographic algorithms, key agreement protocols, the capacity information, etc.
Key Agreement Module (KAM)	The software running in the eUICC to perform the key agreement protocol to agree SM-DP Credentials with SM-DP
Device Module (DM)	The software running in the Device to perform the roles of the intermediary between eUICC and SM-SR
Profile Installer (PI)	The software running in the eUICC to perform the decryption and verification of the protected profile blocks from SM-DP and provision the profile blocks
Profile Manager (PM)	The software running in the eUICC to manage profiles as blocks of encrypted data and has relationship with SM-SR

B. Secure Profile Provisioning Architecture (SPA)

SPA mainly consists of two procedures; Pre-Provisioning Procedure and Secure Profile Provisioning Procedure.

1) Pre-Provisioning Procedure

SPA involves the pre-provisioning procedure, conducted during the eUICC-based device manufacturing. Through this procedure, SM can collect the necessary information for SPA, and the verification of the capability information of the eUICC can be possible when performing the secure profile provisioning in the field. The detail of this pre-provisioning procedure is depicted in Fig.3.

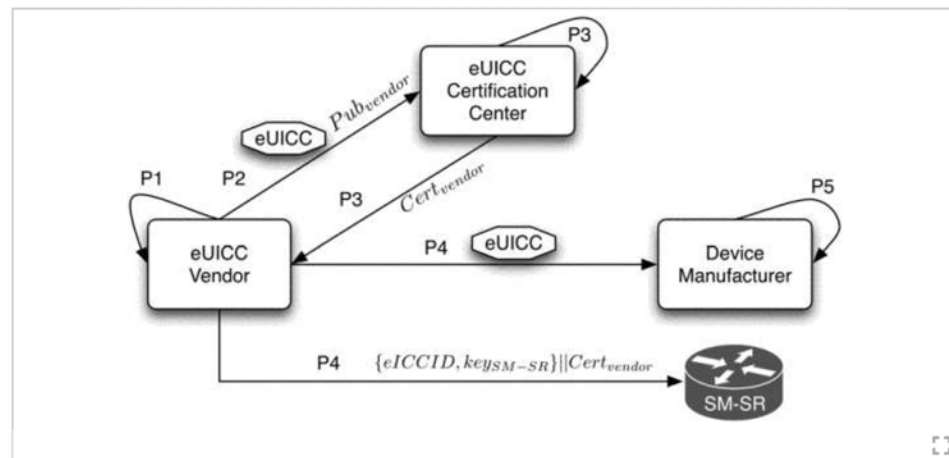


Fig. 3. Pre-Provisioning Procedure

P1 The eUICC Vendor develops eUICCs. During this, the SM-SR Credentials ($keys_{SM-SR}$) and the capability information ($CI_{installed}$) of each eUICC are installed into each eUICC. Then, the eUICC Vendor generates the public key pair, and installs the private key ($Priv_{vendor}$) into the eUICCs.

P2 The eUICC Vendor requests the evaluation of the eUICCs to the eUICC Certification Center with the public key (Pub_{vendor}) of Step P1.

P3 Only after the successful evaluation, the eUICC Certification Center responds the results with the certificate ($Cert_{vendor}$) by using Pub_{vendor} .

P4 The eUICC Vendor can install the received certificate into the eUICCs, and sends the developed and evaluated eUICCs to the device manufacturers. Simultaneously, the eUICC Vendor sends the information to SM-SR required for SPA such as the certificate, the eUICC IDs, the SM-SR Credentials, the mapping table between eUICC IDs and SM-SR Credential indexes, etc. in the secure manner.

P5 The Device Manufacturer develops the eUICC-based devices using the received eUICCs.

2) Secure Profile Provisioning Procedure

After the pre-provisioning procedure, the eUICC-installed devices are deployed in the field. For their operations, the profiles should be provisioned to the eUICCs. Fig. 4 depicts the secure profile provisioning procedure in detail. For the simplicity, we only denote one eUICC and one SM (one SM-SR and one SM-DP); however, the number of the entities can be extended depending on the needs of the market.

P1 The eUICC initially accesses SM-SR using the Provisioning Profile. In addition, the eUICC can access SM-SR via WiFi, USB, etc. without the Provisioning Profile. Also, the provisioned profile for accessing the mobile network can be utilized to reach to SM-SR.

P2 After the establishment of the communication channel, the eUICC sends the message including the eUICC ID ($eICCID$) and its capability information ($CI_{proposal}$) to SM-SR for the capability negotiation. The eUICC ID and capability information are signed by the eUICC Vendors private key, and this signed value needs to be added to the message. If there is no KAM installed onto the eUICC, the eUICC can replace the key agreement protocol of the CI message with proper value such as null, blank, error message, etc. to inform SM that the installation is required. Simultaneously, the eUICC can also send the eUICC Vendors certificate if necessary.

P3 SM acquires the eUICC Vendors certificate from the eUICC Certification Center or the message sent by the eUICC in Step 2. With that certificate, SM-DP verifies the signature of the eUICC ID and the capability information.

P4 Then, SM-DP sends the message including the selected capabilities ($CI_{selected}$), the cryptographic algorithms, the key agreement protocol, the secure communication protocol and so forth) and its hashed value to the eUICC via SM-SR if the decision is completed based on the received capability information. The capability information about the secure communication protocol can be chosen through the interworking between SM-SR and SM-DP.

P5 PM of eUICC and the SM-SR establish the secure communication channel based on the selected protocol. The SM-SR Credential can be same as or derived from the master secret installed in Step P1.

P5 The blocks of KAM from SM-DP and the acknowledgement from the eUICC can be sent through the secure communication channel established in Step 5 and installed to the eUICC if the message of Step 2 indicates that the installation is necessary. This *can* be happened when the change of security policy is required, the vulnerabilities of the cryptographic algorithm used in that module are found, or other reasons are happened.

P6 KAM of eUICC and SM-DP perform the key agreement protocol dynamically to generate the SM-DP Credentials (key_{SM-DP}) through the established secure communication channel between the eUICC and SM-SR.

P7 Using the agreed SM-DP Credentials and the selected cryptographic algorithms, SM-DP encrypts each block of profile ($Profile_i$) and generates its MACed value. Then, SM-DP sends the protected profile blocks to the eUICC via the secure communication channel of SM-SR. PM of eUICC performs the necessary cryptographic operations for the secure communication channel, and then PI of eUICC decrypts and verifies the received profile blocks. If no problem found, the profile blocks are provisioned inside the eUICC by PI. Then, PI sends the acknowledgement for each block of profile with its MACed value to SM-DP. This step continues recursively until the provisioning is completed.

C. Proposed Internal Architecture of eUICC

The change of the internal architecture of the eUICC compared to the conceptual architecture in [2] is inevitable to support SPA as depicted in Fig. 5. Since new module, KAM is added to the internal architecture of eUICC, interworking among modules should be considered to reflect this change. DM of the device should be implemented to handle the messages from SM-SR. Handled messages are sent to PI for provisioning, and PI interworks with KAM to retrieve the agreed SM-DP Credentials for decrypting and verifying the protected profile blocks. After completing of cryptographic operations, PI provisions the profile into the eUICC.

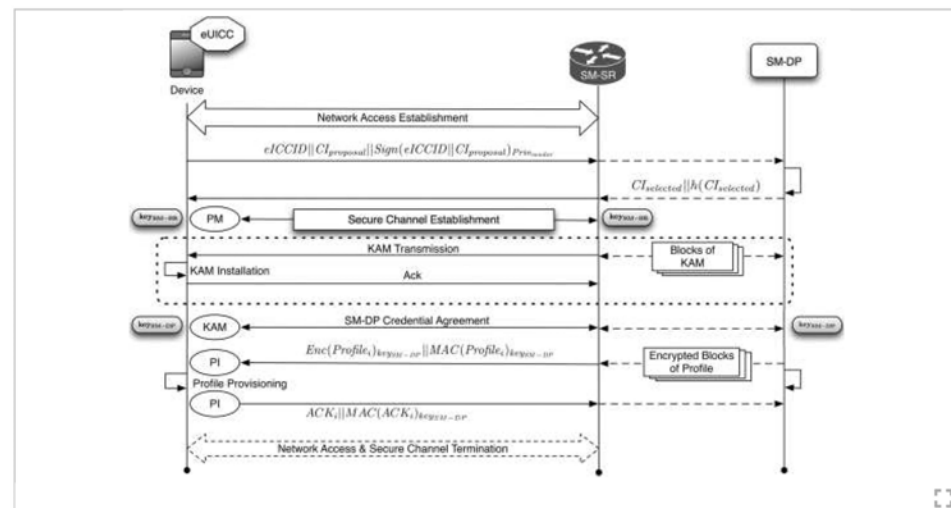


Fig. 4. Secure Profile Provisioning Procedure

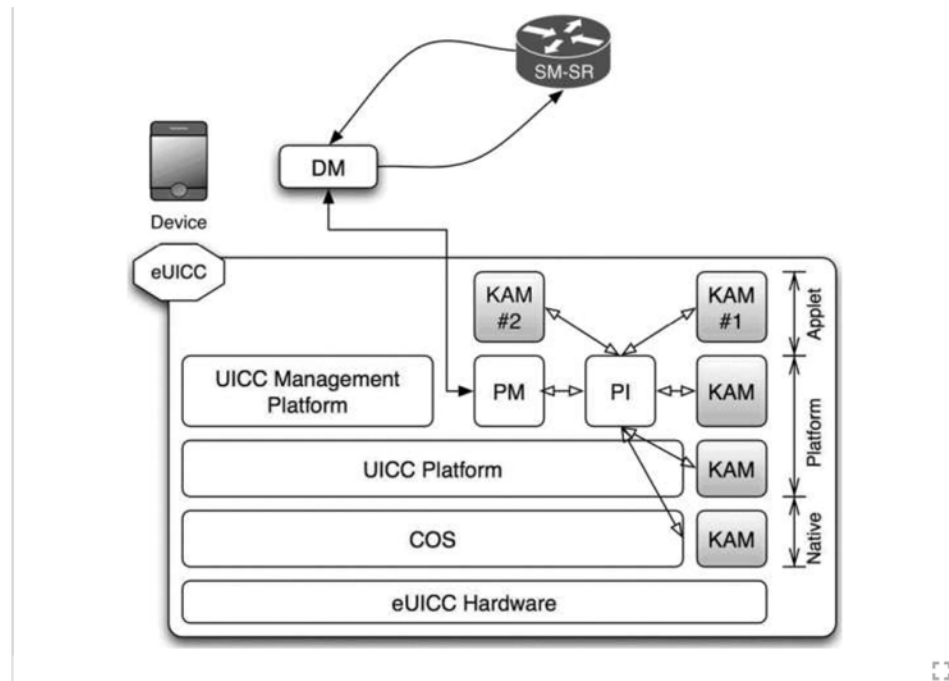


Fig. 5. Internal Architecture of eUICC

1) Key Agreement Module (KAM)

KAM is software running inside the eUICC to perform the key agreement protocol, which is already known by eUICC and SM-DP after the capability negotiation of SPA, with SM-DP. KAM is mainly utilized to agree the SM-DP Credentials on demand. For the security, the SM-DP Credentials should not be revealed to *any* party, even SM-SR, except for eUICC. To accomplish this, KAM is designed and applied to SPA. KAM can provide the asymmetric algorithm-based key agreement protocol, or possess the master secrets of SM-DP Credentials, which are accessible only after the authentication and where SM-DP Credentials are derived afterward.

KAM can be different from each SM-DP if the capabilities of the eUICC are affordable. That is, depending on the security policies, the security threats found in some specific algorithms, and other reasons, different KAMs can be installed. Depending on the selected cryptographic algorithms for protecting profiles, KAM should be able to agree all kinds of SM-DP Credentials. KAM can be various forms like installable applet, native program of COS (Chip OS) or component of the UICC platform (e.g. Javacard Platform or GP) as shown in Fig. 5.

2) Capability Information

Depending on the hardware specification and the implementations, the eUICC can be different from each other with respect to supporting cryptographic algorithms, affordable key agreement protocols, affordable secure communication protocols, total and available memory capacities, existences of KAM or related information, etc. If SM should manage all of this information, necessary storage can be drastic. Then, this complexity yields the increment of error and management cost. Therefore, this capability information should be stored inside the eUICC and be retrieved by the external entities if necessary. Without the guarantee of the correctness and the verification about this information, SM has no method to convince that the capability of the eUICC is correctly implemented. Therefore, the technical method to verify the capability information sent from the eUICC should be existed. To address these issues, we propose a methodology to store the capability information of each eUICC inside itself and verify this information if necessary.

SECTION V.

Analysis

In this section, we evaluate our SPA against the requirements defined in [Section III](#). Furthermore, we compare our SPA with the approach mentioned in [Section II](#).

A. Security Features

SPA provides the secure profile provisioning together with supporting the non-exposure of key and the verification of eUICC's information (capability information).

Mutual Authentication - Depending on the key agreement protocol supported by KAM, the mutual authentication between eUICC and SM-DP can be provided or not. However, there might exist *no* key agreement protocol without mutual authentication.

Confidentiality - Confidentiality is guaranteed since the profile being transmitted is encrypted by the SM-DP's credential.

Data Integrity - Integrity is guaranteed by performing MAC operation for all blocks of profile by SM-DP and all acknowledgements by the eUICC. Therefore, eUICC and SM-DP ensure that data being received has not been altered.

Data Origin Authentication - Data origin authentication is guaranteed by performing MAC operation using the agreed SM-DP Credential for all blocks of profile by SM-DP and all acknowledgements by the eUICC. Therefore, eUICC and SM-DP ensure that all data actually came from SM-DP and eUICC, respectively.

Non-Exposure of Key - Because KAM embeds the shared secret onto the eUICC or performs the key agreement protocol with SM-DP, any SM-DP Credential cannot be shared with other entities.

Verification of Capability Information - Since the capability information of eUICC is signed by eUICC Vendors private key stored inside the eUICC, SMs can ensure that the received capability information was evaluated successfully by eUICC Certification Center and has not been altered.

B. Cryptographic Performance of eUICC

To support SPA, the overhead of the eUICC can be increased since the eUICC should perform additional cryptographic operations. the cryptographic performance can be evaluated by measuring the time required for the eUICC to perform necessary cryptographic operations. Therefore, the overall cryptographic performance is calculated as follows:

$$P(A) = \sum t(f) \quad (1)$$

[View Source](#)

where $P(A)$, is the overall time for the eUICC to perform A and $t(f)$ is the measured time for the eUICC to perform specific cryptographic operation, f .

Using (1), $P(SPA)$ can be calculated as follows:

$$P(SPA) = P(SCP) + P(KAP) + N\{t(Dec) + t(MAC)\} + t(Sign) + t(h) \quad (2)$$

[View Source](#)

where N is number of blocks of profile, SCP is Secure Communication Protocol and KAP is Key Agreement Protocol.

If the cryptography algorithm for profile decryption is the symmetric and identical to GP SCP, $P(SCP)$ can be approximated to (3) since secure communication protocols mostly include key agreement and protection of blocks of data with verification of MAC values.

$$P(SCP) \approx P(KAP) + N\{t(Dec_{sym}) + t(MAC)\} \quad (3)$$

View Source

Then, (2) can be simplified as follows:

$$P(SPA) \approx 2P(SCP) + t(Sign) + t(h) \quad (4)$$

View Source

For better understanding, we compare it with the performance of GP SCP (t) in Section II. The cryptographic performance of GP SCP to be performed by eUICC can be as follows:

$$P(GP) = t = 2P(SCP) \quad (5)$$

View Source

By subtracting (5) from (4), we can evaluate the performance increment due to SPA in the case when the symmetric algorithm is utilized for the profile protection. Then, we can conclude that the cryptographic overhead increased by SPA can be as follows:

$$\sigma_{sym} \approx t(Sign) + t(h) \quad (6)$$

View Source

where σ_{sym} is the performance difference of the eUICC when the symmetric algorithm is applied.

Suppose that the asymmetric algorithm is utilized for protecting the blocks of profile. Even *though* not suitable for the large block of profile, the performance review against every possible candidates can be meaningful for new ecosystem. To estimate the overhead of the asymmetric case, we subtract (5) from (2) directly and assign (3), then we acquire following (7):

$$\sigma_{asym} \approx M\{t(Dec_{asym} + t(MAC))\} - N\{t(Dec_{sym} + t(MAC))\} + t(Sign) + t(h) \quad (7)$$

View Source

where σ_{asym} is the performance difference of the eUICC when the asymmetric algorithm is applied, M is the number of blocks of profile (length=L) for the asymmetric algorithm (block length=b1) and N is the number of blocks of same profile for the symmetric algorithm (block length=b2).

Since the block lengths of the symmetric and the asymmetric are different each other, and ciphering and MACing are processed for each block of profile, (7) can be expressed as follows using (6):

$$\sigma_{asym} \approx \sigma_{sym} + \left\lceil \frac{L}{b1} \right\rceil \cdot t(Dec_{asym}) - \left\lceil \frac{L}{b2} \right\rceil \cdot t(Dec_{sym}) + \left\{ \left\lceil \frac{L}{b1} \right\rceil - \left\lceil \frac{L}{b2} \right\rceil \right\} \cdot t(MAC) \quad (8)$$

View Source

where $\lceil x \rceil$ is the ceiling function of x .

For the reference, according to the cryptographic performance of the CCP (Crypto Co-Processor) used for (e)UICC [14], we calculate the approximate value of each σ against the profile length, L , as shown in Fig. 6. We consider two cases to estimate σ_{asym} depending on the type of MAC algorithm; CMAC (Cipher-based MAC) or HMAC (Hash-based MAC). As the profile length, L , is increased, all σ_{asym} values are increased linearly. Therefore, for the better performance, the symmetric algorithm for encrypting and decrypting the profile is preferable since the constant overhead (σ_{sym}) is promised.

C. Comparison

We compare SPA with the approach of the GP SCP with respect to the security, the performance and the drawbacks mentioned in II. Besides the improved security, SPA overcomes all drawbacks of GP SCP with reasonable overhead, σ of the performance as shown in Table II.

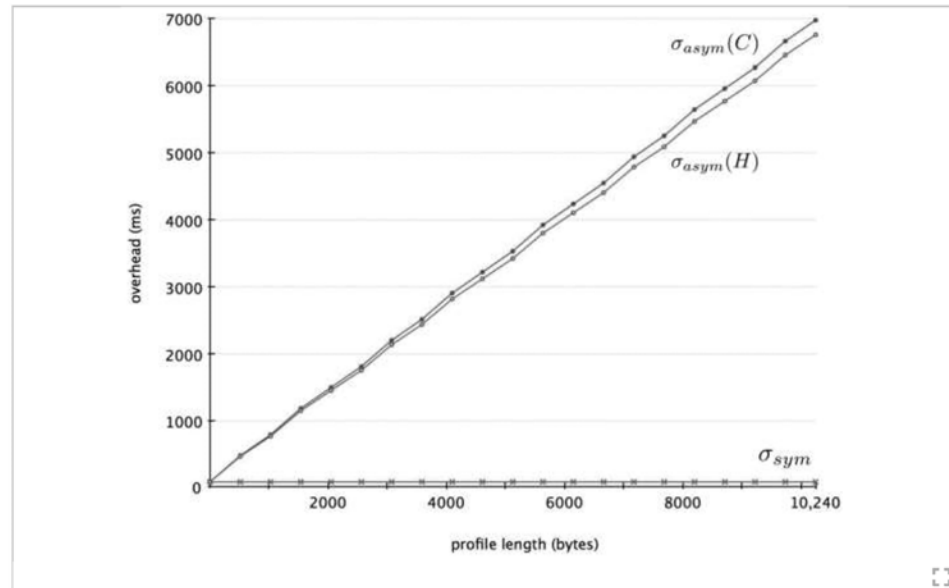


Fig. 6. Cryptography Performance of eUICC

SPA does not necessitate the pre-shared secret between SM-SR and SM-DP since SM-DP Credential is *generated* on demand. As this inconvenient procedure is removed, the scalability is improved. SM-DPs *do not have* to implement the communication facilities to be accessed by eUICCs directly since one communication channel and one SCP session with SM-SR are required. This can reduce the management cost of eUICC systems and the network overhead of eUICCs. By utilizing KAM and the capability information of eUICC, SPA supports various kinds of *security* protocols if eUICC is affordable. Therefore, diverse key agreement protocols, secure communication protocols and cryptographic algorithms can be applied depending on the circumstances around the eUICC.

TABLE II. COMPARISON

	Criteria	GP SCP	SPA
Security	Mutual Authentication	0	o
	Confidentiality	0	0
	Data Integrity	0	0
	Data Origin Authentication	0	0
	Non-Exposure of Key	$\triangle$	0
	Verification of Capability Information	$\times$	0
Performance	Communication Channel	2	1
	Secure Communication Channel	2	1
	Cryptographic Overhead of (e)UICC	t	$t + \sigma$

	Criteria	GP SCP	SPA
General	Scalability	△	0
	Efficiency	△	0
	Flexibility	△	0

SECTION VI.

Conclusion and Further Works

In this paper, we propose a novel secure profile provisioning architecture, SPA for the eUICC. For the profile provisioning, we primarily defines the overall architecture of the secure profile provisioning for eUICC. To protect the profile and overcome the drawbacks of current UICC technology, we utilize the capability negotiation using the capability information of eUICC and the special module, KAM, to agree SM-DP Credentials dynamically during the profile provisioning procedures. To show the excellence of SPA, we analyze our SPA against security, cryptographic performance of eUICC, scalability, efficiency and flexibility. Our analysis depicts that SPA is superior to the methodology to apply GP SCP with reasonable overheads of eUICC if the symmetric algorithm for protecting profile is applied.

As further works, we intend to study about new secure communication protocol and key agreement protocol optimized for eUICC systems. This study should mainly deal with new circumstances raised by the eUICC provisioning ecosystem such as the inevitable remote management of numerous eUICCs, the security concerns for all related entities around, etc.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

Authors	▼
Figures	▼
References	▼
Citations	▼
Keywords	▼
Metrics	▼

[Back to Results](#)

IEEE Personal Account

CHANGE
USERNAME/PASSWORD

Purchase Details

PAYMENT OPTIONS
VIEW PURCHASED
DOCUMENTS

Profile Information

COMMUNICATIONS
PREFERENCES
PROFESSION AND
EDUCATION
TECHNICAL INTERESTS

Need Help?

US & CANADA: +1 800
478 4333
WORLDWIDE: +1 732
981 0060
CONTACT & SUPPORT

Follow

f @ in v

11/12/25, 3:51 PM

Secure Profile Provisioning Architecture for Embedded UICC | IEEE Conference Publication | IEEE Xplore

[About IEEE Xplore](#) | [Contact Us](#) | [Help](#) | [Accessibility](#) | [Terms of Use](#) | [Nondiscrimination Policy](#) | [IEEE Ethics Reporting](#)  | [Sitemap](#) | [IEEE Privacy Policy](#)

A public charity, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

© Copyright 2025 IEEE - All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies.

EXHIBIT B

A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications

Pierre E. ABI-CHAR, Abdallah MHAMED
UMR CNRS 5157
GET/Institut National des Télécommunications
9 rue C. Fourier - 91011 Evry CEDEX - France
{pierre.abichar; abdallah.mhamed}@int-edu.eu

Bachar EL-HASSAN
Libanese University
Faculty of Engineering
Tripoli - Lebanon
bachar_elhassan@ul.edu.lb

Abstract

The increasing progress in wireless mobile communication has attracted an important amount of attention on the security issue. To provide secure communication for mobile devices, authenticated key agreement protocol is an important primitive for establishing session key. So far, several protocols have been proposed to provide robust mutual authentication and key establishment for wireless local area network (WLAN). In this paper we present a fast and Secure Authenticated Key Agreement (EC-SAKA) protocol based on Elliptic Curve Cryptography. Our proposed protocol provides secure mutual authentication, key establishment and key confirmation over an untrusted network. The new protocol achieves many of the required security and performance properties. It can resist dictionary attacks mounted by either passive or active networks intruders. It can resist Man-In-The Middle attack. It also offers perfect forward secrecy which protects past sessions and passwords against future compromise. In addition, it can resist known-key and resilience to server attack. Our proposed protocol uses ElGamal signature techniques (ECEGS). We show that our protocol meets the above security attributes under the assumption that the elliptic curve discrete logarithm problem is secure. Our proposed protocol offers significantly improved performance in computational and communication load over comparably many authenticated key agreement protocols such as B-SPEKE, SRP, AMP, PAK-RY, PAK-X, SKA, LR-AKE and EC-SRP.

1 Introduction

In key agreement protocol two or more distributed entities need to share some key in secret, called session key. This session key can then be used to achieve some cryptographic goal such as confidential communication chan-

nel between entities or data integrity. There are two kinds of key establishment protocols: Key transport protocols in which a key is created by one entity and securely transmitted to the second entity, and Key agreement protocols in which both parties contribute information which jointly establish the shared key [14]. A key agreement protocol is said to provide implicit key authentication if entity A is assured that no other entity aside from a specifically identified second entity B can possibly learn the value of a particular secret key. A key agreement protocol which provides implicit key authentication to both entities is called an authenticated key agreement protocol. If both implicit key authentication and key confirmation are provided, then the key establishment protocol is said to provide explicit key authentication. A key agreement protocol which provides explicit key authentication to both entities is called an authenticated key agreement with key confirmation [14]. The security of Elliptic Curve cryptography relies on the discrete logarithm problem over the points on an elliptic curve. The best known methods to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) are Pollard approach and Pohlig-Hellman method. They are fully exponential while the best known methods to solve the Integer Factorization Problem (IFP) and the Discrete Logarithm Problem (DLP), on which most of the non-ECC cryptosystems rely, are sub-exponential. In fact, ECC can significantly reduce the computation and storage overhead.

In this paper we present a fast and secure three-pass authenticated key establishment protocol for low power mobile wireless devices that provides secure mutual authentication and key agreement with key confirmation. The EC-SAKA (Secure Authenticated Key Agreement) is based on the Elliptic Curve Cryptography [19], on the EC ElGamal Signature Scheme (ECEGS), on SKA (Simple Key Agreement) protocol [17] and on the assumption that the ECC discrete logarithm problem is secure [19]. Our proposed protocol achieves many of desirable security requirements

and performances.

The ECEGS is used to make the proposed protocol fast and efficient, both from a computational point of view and in the amount of information that needs to be exchanged in the scheme. It is also used to minimize the amount of computational performed by Alice, in particular. This is desirable because, in many practical applications, Alice's computations will be performed by a smart card with low computing power, while Bob's computations will be performed by a more powerful computer.

The protocol described in this paper establish a shared key K between the two entities. A key derivation function should then be used to derive the session key K_s based on the shared key.

The remainder of this paper is organized as follows. Section 2 reviews the desirable properties needed for WLAN (Wireless Local Area Network) authenticated protocols. Section 3 presents the overall architecture of our proposed protocol. In section 4, the security analysis is described. In section 5, a complete comparison over comparably many protocols is listed. Finally, section 6 makes concluding remarks.

2 DESIRABLE PROPERTIES FOR KEY AGREEMENT PROTOCOLS

A number of desirable properties for key agreement protocols have been identified [2] and nowadays most of the protocols are analyzed using these properties which are described below:

- Known-key security: Each run of a key agreement protocol between two entities A and B should produce a unique shared secret key called session key K_s . A protocol should still achieve its goal in the face of an adversary who has learned some other session key.

- Perfect forward secrecy: If long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities is not affected.

- Key-compromise impersonation: Suppose that A's long-term private key is disclosed. Clearly an adversary that knows this value can now impersonate A, since it is precisely this value that identifies A. However, it may be desirable that this loss does not enable an adversary to impersonate other entities to A.

- Unknown key-share: Entity A cannot be coerced into sharing a key with entity B without A's knowledge, i.e., when A believes the key is shared with some entity $C \neq B$, and B (correctly) believes the key is shared with A.

- Key control: No other entity should be able to force the session key to a preselected value.

In addition, Identification protocols should have other properties which are related to performance. Because round

trips and large blocks are critical factors in terms of communication load and because exponentiations and random numbers are to be critical factors in terms of computation load, such properties are listed below:

- Computational efficiency: this includes the number of operations required to execute a protocol. In order to achieve this property, the protocol should have the minimum number of operation as possible.

- Communication efficiency: This includes the number of passes (message exchanges) and the bandwidth required (total number of bits transmitted).

Other desirable properties are:

- Nature of security guarantees: including provable security and zero-knowledge properties.

- Storage of secrets: This refer to the location and the method used (e.g., software only, local disks, hardware tokens, etc.) to store critical keying material.

3 Our proposed Protocol

In this section we describe the EC-SAKA protocol in which two entities are both proving their identities to each other and establish a common session key in order to elaborate a secure connection. Alice and Bob represent a client and a server respectively.

3.1 EC-SAKA Protocol Parameters

Many researchers have examined elliptic curve cryptosystems, which were firstly proposed by Miller [15] and Koblitz [8]. The elliptic curves which are based on the elliptic curve discrete logarithm problem over a finite field have some advantages than other systems: the key size can be much smaller than the other schemes since only exponential-time attacks have been known so far if the curve is carefully chosen [9], and the elliptic curve discrete logarithms might be still intractable even if factoring and the multiplicative group discrete logarithm are broken. In this paper we use an elliptic curve E defined over a finite field F_p . The elliptic curve parameters to be selected [12] and [11] are:

- 1 -Two field elements a and $b \in F_p$, which define the equation of the elliptic curve E over F_p (i.e., $y^2 = x^3 + ax + b$ in the case $p \geq 4$, where $4a^3 + 27b^2 \neq 0$).

- 2 -Two field elements x_p and y_p in F_p , which define a finite point $P(x_p, y_p)$ of prime order in $E(F_p)$ (P is not equal to O , where O denotes the point at infinity).

- 3 -The order n of the point P .

The Elliptic Curve domain parameter can be verified to meet the following requirements [12] and [11]. In order to avoid the Pollard-rho [16] and Pohling-Hellman algorithms for the elliptic curve discrete logarithm problem,

it is necessary that the number of F_p -rational points on E , denoted by $\#E(F_p)$, be divisible by a sufficiently large prime n . To avoid the reduction algorithms of Menezes, Okamoto and Vanstone [1] and Frey and Ruck [5], the curve should be non-supersingular (i.e., p should not divide $(p+1 - \#E(F_p))$). To avoid the attack of Semaev [18] on F_p -anomalous curves, the curve should not be F_p -anomalous (i.e., $\#E(F_p) \neq p$).

The table below (Table 1) shows the ECC mathematical parameters that are used in our proposed protocol.

Table 1. EC Mathematical Notations

Index	Explanation
$(h())$	one-way hash function
p, q	large prime numbers, where $p = 2 \cdot q + 1$
P, Q	Random points over elliptic curve
a, b	Random generated private keys
A, B	Agreed public key
E	non-supersingular elliptic curve
B	$B \in E(F_q)$ with order q
$x(Q)$	x coordinate of point Q
x_u	user 'u' secret key
Y_u	u's public key with $Y_u = (x_u B) \bmod p$
pw	The user's password

In the following, we will give an introduction to the EC-discrete logarithm problem, to Diffie-Hellman key exchange based on EC and to the elliptic curve based ElGamal signature scheme (ECEGS), which are the core theories used in our proposed protocol.

Let E be an elliptic curve defined over a finite field F_p and let $P \in E(F_p)$ be a point of order n . Given Q where $Q \in E(F_q)$, the elliptic curve discrete logarithm problem is to find the integer l , $0 \leq l \leq n-1$, such that $Q = l \cdot P$.

The Diffie-Hellman key agreement protocol runs as follows: Alice selects a random number n_a and computes $Y_a = n_a B$, he sends Y_a to Bob. Similarly Bob computes $Y_b = n_b B$ and sends Y_b to Alice. Alice and Bob generate the same key $K = n_a Y_b B = n_b Y_a B = n_a n_b B$.

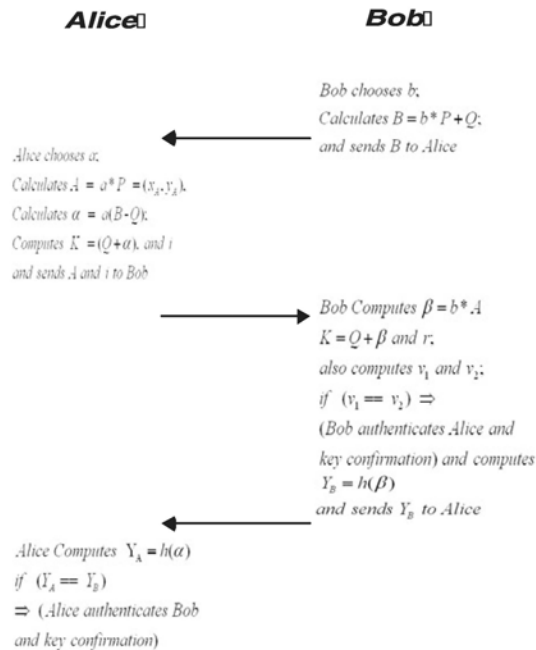
The ECEGS runs as follows: Alice selects a random number x_a , where $2 \leq x_a \leq n-2$, as his secret key and computes the corresponding public key $Y_a = x_a B$. Therefore the public key and the private key are (E, Y_a, B, n) and x_a . To generate a signature for a message m , Alice will select a random number k , where $2 \leq k \leq n-2$ computes $R = kB$ and computes $r = x(KB) \bmod n$. If $r \neq 0$, then computes $s = K^{-1}(h(m) + x_a r) \bmod n$. The couple (R, s)

will be Alice's signature of m . To verify the signature Bob will first confirm that r and $s \in [2, n-2]$ and then computes $v_1 = sR$ and $v_2 = h(m)B + rY_a$. Finally Bob will accept the signature if and only if $(v_1 == v_2)$.

3.2 EC-SAKA Protocol Description

The main goal of our proposed protocol is to achieve mutual authentication and key confirmation between the client and the server in order to establish a secure channel. Our proposed applies the ECEGS to the SKA protocol to enhance the safely level and protocol simplification in terms of computational and communications load. Our proposed protocol consists of three flows and it is illustrated in Figure 1.

Figure 1. The EC-SAKA Protocol Scheme



Before running the authentication procedure, the client select an elliptic curve $E(Z_p)$ defined on Z_p . Alice, select a random point P over the elliptic curve with order n . n is a large prime number. Alice chooses a password pw , computes $x = h(pw)$ and calculates Q where $Q = x * P$. In addition, Alice generates strong number p and q where $p = 2 * q + 1$. Once the following parameters (E, Q, P, p, q, pw) are generated, Alice transfers (E, Q, P, n) to the server, Bob, in a secure way. Once

these steps are done, the session key generation procedure will be executed as follow:

Within the first flow, Bob chooses a random challenge b , where $1 \leq b \leq n-1$, then he calculates the point B where

$$B = b * P + Q \quad (1)$$

Finally he sends B to Alice.

Within the second flow, Alice chooses a random challenge a , where $1 \leq a \leq n-1$, then computes A where

$$A = a * P = (x_A, y_A) \quad (2)$$

and calculates α where

$$\alpha = a(B - Q) \quad (3)$$

and $K = Q + \alpha$. In addition, Alice calculates $r = (x_A) \bmod(n)$ and checks that $r \neq 0$. If so, then Alice computes

$$i = a^{-1}(h(\alpha) + x * r) \bmod(n) \quad (4)$$

Finally (A, i) becomes the signatures pair and Alice transfers A and i to the server.

Within the third flow, Bob computes

$$\beta = b * A \quad (5)$$

Computes $K = Q + \beta$ and computes $r = x_A \bmod n$. To verify the signature, first Bob checks that $(r, i) \in [2, n-2]$. If so, then Bob computes

$$v_1 = i * A \quad (6)$$

and calculates

$$v_2 = (h(\beta)P) + r * Q \quad (7)$$

Finally, Bob checks if $(v_1 == v_2)$, if so, Bob authenticates Alice and Bob can be confirmed that Alice has actually established the same shared session key. Then Bob computes:

$$Y_B = h(\beta) \quad (8)$$

and finally he sends Y_B to Alice.

In order to authenticate Bob, Alice will compute:

$$Y_A = h(\alpha) \quad (9)$$

and then Alice will verify the value of Y_A by checking that $(Y_A == Y_B)$, if so, if they match, then Alice authenticates Bob and Alice can be confirmed that Bob has actually established the same shared session key with her.

Finally, Alice and Bob agree on the common session key K_s where

$$K_s = h(ID(Alice) || ID(Bob) || K) \quad (10)$$

Both sides will agree on the session Key K_s if all steps are executed correctly. Once the protocol run completes successfully, both parties may use K_s to encrypt subsequent session traffic in order to create a confidential communication channel.

4 Security Analysis

In the following section, we will analyze the security of our proposed protocol.

The EC-SAKA protocol is considered to be a secure authenticated key establishment protocol, if it satisfies the following properties:

-Passive attack: Suppose that Oscar the attacker perform a passive attack, then the session will terminate with both parties accepting. That is, Bob and Alice successfully identify themselves to each other, and they both compute the session key. So, Oscar, the adversary, cannot compute any information about the common shared session key K_s by assuming the intractability of the elliptic curve discrete algorithms problem. Therefore the EC-SAKA protocol resists against the passive attack.

-Man in the middle attack (or active attack): Suppose that an attacker, Oscar, intercepts B and replaces it with B' , Oscar then receives A and i from Alice. He would like to replace i by i' , as before. However, this means that he must calculates α where $\alpha = a(B' - Q)$ but unfortunately for Oscar, he can not compute the value of α because he does not know the the value of Q neither the value of a . So, Oscar will not be able to compute K , neither K_s . Therefore the EC-SAKA protocol thwarts the man in-the-middle attack.

-Dictionary attack: In dictionary attack, the attacker finds the real password by repeating a process of guessing the password of legal client and applying the passwords. The dictionary could be performed in offline or online mode. In our proposed protocol, it is impossible to get the real password since a one way hash function is applied to the password and during the protocol process, the shared Key K used in the calculation of K_s is calculated from b and a which are generated every new session; and by assuming the intactability of elliptic curve discrete logarithm problem Therefore the EC-SAKA protocol thwarts the offline and online dictionary attack.

-Known-key attack: In this attack, an adversary will capture the session key from an eavesdropped session. In our proposed protocol, the client and the server both generates new b and a every new session, and in addition the shared key K is generated with every new session also. Thus our proposed protocol is secure against known key attacks assuming that the elliptic discrete problem is intractable.

-Perfect forward secrecy: The perfect forward secrecy is that an exposed password does not enable an attacker to derive session keys of past communication sessions. In our protocol, the security of perfect forward secrecy is based upon the assumption that the elliptic curve discrete problem is intractable and on the value of the key K . Even if the attacker knew the correct password, the attacker still cannot compute the previous session keys because K_s is derived from the shared key K which is generated from the value of a and b . Therefore, the EC-SAKA protocol satisfies the property of perfect forward secrecy.

-Resilience to server compromised: if the host's password file is compromised, an adversary can not use it to impersonate legitimate user since the server does not store the file of the password, instead the value of Q where $Q = x * P$ is stored. Thus, Oscar has to solve the elliptic curve discrete logarithm problem in order to retrieve the value of password. The EC-SAKA protocol provides resilience to server compromise.

5 Performance evaluation: Efficiency and Comparison:

Computation cost and communication cost are the most important aspects of password authentication protocols which affect the overall performance. They include number of steps, exponentiations, large blocks, symmetric encryption and decryption, hash functions and random numbers. In this section, we compare the EC-SAKA protocol with the following protocols: Leakage-Resilient Authenticated Key Exchange (LR-AKE) protocol, Simple Key Agreement (SKA) protocol, Secure Remote Password (SRP) protocol, EC-SRP, Simple Password Exponential Key Exchange (B-SPEKE) protocol, Password-Authenticated Key Exchange (PAK-X and PAK-RY) protocols and Authentication Memorable Password (AMP) protocol. The comparison is done in terms of number of steps, random numbers, exponentiations and hash functions. Table 2 shows the compared result for number of steps and exponentiation. Table 3 shows the compared result for random numbers and hash functions numbers.

It is clear from Table 2 that the EC-SAKA protocol has the minimal cost in terms of number of steps and exponenti-

Table 2. Comparison of Performance-1-

Protocol	Rounds	Exponentiations		
		Client	Server	Total
B-SPEKE	4	3	4	7
SRP	4	3	3	6
AMP	4	2	3	5
PAK-RY	3	5	4	9
PAK-X	3	5	4	9
SKA	3	2	3	5
LR-AKE	3	3	2	5
AKEECC	4	2	2	4
EC-SRP	3	2	2	4
EC-SAKA	3	1	0	1

ations compared with these above protocols. We can easily notice that B-SPEKE [4], SRP [20], AKEECC [7] and AMP [10] require 4 rounds while PAK-RY [3], PAK-X [13], SKA [17], LR-AKE [6] and EC-SAKA require 3 rounds. In addition, the computational load was clearly improved using EC-SAKA protocol because, as noted in table 2, EC-SAKA requires 1 exponentiations, one for the client and nothing for the server. While for the other protocols, including SKA, LR-AKE, AKEECC [7] and EC-SRP [7], they require at least 4 exponentiations.

Table 3. Comparison of Performance-2-

Protocol	Random N.	Hash Function N.
SRP	2	6
AMP	2	9
PAK-RY	3	8
PAK-X	3	10
SKA	2	7
LR-AKE	2/4	6
AKEECC	2	6
EC-SRP	3	5
EC-SAKA	2	5

From Table 3, we can easily notice that the EC-SAKA protocol requires 2 random numbers and 5 hash functions while all the other protocols require more. In addition, for the EC-SRP and AKEECC protocols described in [7], we can easily notice that our protocol is better then these two protocols in terms of hash functions numbers. For the EC-SRP protocols described in [7], EC-SRP protocol was proposed for a one way authentication while our proposed protocol, EC-SAKA, provides mutual authentication.

6 Conclusion

Wireless network access are nowadays very important for users. This paper describes a new network access mech-

anism for wireless local area networks. In this paper, we introduce a fast and secure authenticated key agreement protocol based on elliptic curve cryptography and provides mutual authentication and explicit key establishment. We also give a formal approach to prove its security based on the previous work by [7]. Our scheme is simple, easy to realize, and secure against both passive and active attacks. It also resists many others attacks as described in section 4. Our proposed protocol is compared to well-known protocols such as B-SPEKE, SRP, EC-SRP, AKEECC, PAK-RY, PAK-X, AMP, SKA and LR-AKE in terms of communication and computation cost and the results were well discussed in the previous section.

Acknowledgments

The authors would like to thanks the following departments RST-INT Evry and FOE-UL for their support and comments. Their suggestions and observations were extremely helpful throughout this paper.

References

- [1] T. O. A. Menezes and S. Vanstone. Reducing elliptic curve logarithms in a finite field. *IEEE Transactions on Information Theory*, vol. 39, pages 1639 – 1646, 1993.
- [2] S. Blake-Wilson, D. Johnson, and A. Menezes. Key agreement protocols and their security analysis. In *Proc. of Sixth IMA International Conference on Cryptography and Coding*, pages 30 – 45. Cirencester, UK, 1997.
- [3] V. Boyko, P. Mackenzie, and S. Patel. Provably secure password authenticated key exchange using diffie-hellman. *EuroCrypt*, pages 156 – 171, 2000.
- [4] D. Jablon. Extended password key exchange protocols immune to dictionary attack. *WETICE Workshop*, pages 248 – 255, 1997.
- [5] G. Frey and H. Ruck. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, vol. 62, pages 865 – 874, 1994.
- [6] I. Hideki, S. Seonghan, and K. Kobara. Authenticated key exchange for wireless security. *IEEE Wireless Communications and Networking Conference*, pages 1180 – 1186, 2005.
- [7] J. K. K. Jung and T. Chung. Password-based independent authentication and key exchange protocol. *Proc. of ICICS-PCM 2003, Singapore*, 2003.
- [8] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, vol. 48., pages 203 – 209, 1987.
- [9] N. Koblitz. Cm-curves with good cryptography properties. *Proc. of Crypto'91, Santa Barbara, USA*, 1992.
- [10] T. Kwon. Ultimate solution to authenticate via memorable password. *Contribution to the IEEE P 1363 Study group for Future PKC Standards*, available for <http://grouper.ieee.org/groups/1363/>.
- [11] M. Q. J. S. L. Law, A. Menezes and S. Vanstone. An efficient protocol for authenticated key agreement. In *Designs, Codes and Cryptography*, vol. 28.
- [12] M. Q. J. S. L. Law, A. Menezes and S. Vanstone. An efficient protocol for authenticated key agreement. *Technical report CORR98-05, Department of CO, University of Waterloo*, 1998.
- [13] P. Mackenzie. More efficient password authenticated key exchange. *CT-RSA*, pages 361 – 377, 2001.
- [14] A. Menezes, P. Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2nd edition, 1996.
- [15] V. Miller. Uses of elliptic curves in cryptography. In *Proc. of Crypto '85, Santa Barbara, USA*, pages 417 – 426, 1986.
- [16] J. Pollard. Monte carlo methods for index computation mod p. *Mathematics of Computation*, vol. 32, pages 918 – 924, 1978.
- [17] E. Ryu, K. Kim, and K. Yoo. A simple key agreement protocol. In *Proc. of IEEE 37th Annual 2003 International Carnahan Conference*, pages 128 – 131, 2003.
- [18] I. Semaev. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. *Mathematics of Computation*, vol. 67, pages 353 – 356, 1998.
- [19] D. R. Stinson. *Cryptography Theory and Practice*. Chapman and Hall/CRC, third edition, 2006.
- [20] T. Wu. Secure remote password protocol. *Interent Symposium on Network and Distribution System Security*, 1998.

A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications

Publisher: IEEE

[Cite This](#) PDFPierre E. ABI-CHAR ; Abdallah MHAMED ; Bachar EL-HASSAN [All Authors](#)

44

Cites in
Papers

20

Cites in
Patents

427

Full
Text Views

Abstract

Document Sections

- 1 Introduction
- 2 Desirable Properties for Key Agreement Protocols
- 3 Our Proposed Protocol
- 4 Security Analysis
- 5 Performance Evaluation: Efficiency and Comparison

[Show Full Outline](#)[Authors](#)[Figures](#)[References](#)[Citations](#)[Keywords](#)[Metrics](#)[More Like This](#)

Abstract:

The increasing progress in wireless mobile communication has attracted an important amount of attention on the security issue. To provide secure communication for mobile devices, authenticated key agreement protocol is an important primitive for establishing session key. So far, several protocols have been proposed to provide robust mutual authentication and key establishment for wireless local area network (WLAN). In this paper we present a fast and secure authenticated key agreement (EC-SAKA) protocol based on elliptic curve cryptography. Our proposed protocol provides secure mutual authentication, key establishment and key confirmation over an untrusted network. The new protocol achieves many of the required security and performance properties. It can resist dictionary attacks mounted by either passive or active networks intruders. It can resist Man-In-The Middle attack. It also offers perfect forward secrecy which protects past sessions and passwords against future compromise. In addition, it can resist known- key and resilience to server attack. Our proposed protocol uses ElGamal signature techniques (ECEGS). We show that our protocol meets the above security attributes under the assumption that the elliptic curve discrete logarithm problem is secure. Our proposed protocol offers significantly improved performance in computational and communication load over comparably many authenticated key agreement protocols such as B-SPEKE, SRP, AMP, PAK-RY, PAK-X, SKA, LR-AKE and EC-SRP.

Published in: The 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007)

Date of Conference: 12-14 September 2007

DOI: 10.1109/NGMAST.2007.4343427

Date Added to IEEE Xplore: 08 October 2007

Publisher: IEEE

► ISBN Information:

Conference Location: Cardiff, UK

▼ ISSN Information:

SECTION 1 Introduction

In key agreement protocol two or more distributed entities need to share some key in secret, called session key. This session key can then be used to achieve some cryptographic goal such as confidential communication channel between entities or data integrity. There are two kinds of key establishment protocols: Key transport protocols in which a key is created by one entity and securely transmitted to the second entity, and Key agreement protocols in which both parties contribute information which jointly establish the shared key [14]. A key agreement protocol is said to provide implicit key authentication if entity A is assured that no other entity aside from a specifically identified second entity B can possibly learn the value of a particular secret key. A key agreement protocol which provides implicit key authentication to both entities is called an authenticated key agreement protocol. If both implicit key authentication and key confirmation are provided, then the key establishment protocol is said to provide explicit key authentication. A key agreement protocol which provides explicit key authentication to both entities is called an authenticated key agreement with key confirmation [14]. The security of Elliptic Curve cryptography relies on the discrete



logarithm problem over the points on an elliptic curve. The best known methods to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) are Pollard approach and Pohlig-Hellman method. They are fully exponential while the best known methods to solve the Integer Factorization Problem (IFP) and the Discrete Logarithm Problem (DLP), on which most of the non-ECC cryptosystems rely, are sub-exponential. In fact, ECC can significantly reduce the computation and storage overhead.

In this paper we present a fast and secure three-pass authenticated key establishment protocol for low power mobile wireless devices that provides secure mutual authentication and key agreement with key confirmation. The EC-SAKA (Secure Authenticated Key Agreement) is based on the Elliptic Curve Cryptography [19], on the EC ElGamal Signature Scheme (ECEGS), on SKA (Simple Key Agreement) protocol [17] and on the assumption that the ECC discrete logarithm problem is secure [19]. Our proposed protocol achieves many of desirable security requirements and performances.

The ECEGS is used to make the proposed protocol fast and efficient, both from a computational point of view and in the amount of information that needs to be exchanged in the scheme. It is also used to minimize the amount of computational performed by Alice, in particular. This is desirable because, in many practical applications, Alice's computations will be performed by a smart card with low computing power, while Bob's computations will be performed by a more powerful computer.

The protocol described in this paper establish a shared key K between the two entities. A key derivation function should then be used to derive the session key K_s based on the shared key.

The remainder of this paper is organized as follows. [Section 2](#) reviews the desirable properties needed for WLAN (Wireless Local Area Network) authenticated protocols. [Section 3](#) presents the overall architecture of our proposed protocol. In [section 4](#), the security analysis is described. In [section 5](#), a complete comparison over comparably many protocols is listed. Finally, [section 6](#) makes concluding remarks.

SECTION 2

Desirable Properties for Key Agreement Protocols

A number of desirable properties for key agreement protocols have been identified [2] and nowadays most of the protocols are analyzed using these properties which are described below:

- **Known-key security:** Each run of a key agreement protocol between two entities A and B should produce a unique shared secret key called session key K_s . A protocol should still achieve its goal in the face of an adversary who has learned some other session key.
- **Perfect forward secrecy:** If long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities is not affected.
- **Key-compromise impersonation:** Suppose that A's long-term private key is disclosed. Clearly an adversary that knows this value can now impersonate A, since it is precisely this value that identifies A. However, it may be desirable that this loss does not enable an adversary to impersonate other entities to A.
- **Unknown key-share:** Entity A cannot be coerced into sharing a key with entity B without A's knowledge, i.e., when A believes the key is shared with some entity $C \neq B$, and B (correctly) believes the key is shared with A.
- **Key control:** No other entity should be able to force the session key to a preselected value.

In addition, Identification protocols should have other properties which are related to performance. Because round trips and large blocks are critical factors in terms of communication load and because exponentiations and random numbers are to be critical factors in terms of computation load, such properties are listed below:

- Computational efficiency: this includes the number of operations required to execute a protocol. In order to achieve this property, the protocol should have the minimum number of operation as possible.
- Communication efficiency: This includes the number of passes (message exchanges) and the bandwidth required (total number of bits transmitted).

Other desirable properties are:

- Nature of security guarantees: including provable security and zero-knowledge properties.
- Storage of secrets: This refer to the location and the method used (e.g., software only, local disks, hardware tokens, etc.) to store critical keying material.

SECTION 3

Our Proposed Protocol

In this section we describe the EC-SAKA protocol in which two entities are both proving their identities to each other and establish a common session key in order to elaborate a secure connection. Alice and Bob represent a client and a server respectively.

3.1 EC-SAKA Protocol Parameters

Many researchers have examined elliptic curve cryptosystems, which were firstly proposed by Miller [15] and Koblitz [8]. The elliptic curves which are based on the elliptic curve discrete logarithm problem over a finite field have some advantages than other systems: the key size can be much smaller than the other schemes since only exponential-time attacks have been known so far if the curve is carefully chosen [9], and the elliptic curve discrete logarithms might be still intractable even if factoring and the multiplicative group discrete logarithm are broken. In this paper we use an elliptic curve E defined over a finite field F_p . The elliptic curve parameters to be selected [12] and [11] are:

- 1 – Two field elements a and $b \in F_p$, which define the equation of the elliptic curve E over F_p (i.e., $y^2 = x^3 + ax + b$ in the case $p \geq 4$, where $4a^3 + 27b^2 \neq 0$).
- 2 – Two field elements x_p and y_p in F_p , which define a finite point $P(x_p, y_p)$ of prime order in $E(F_p)$ (P is not equal to O , where O denotes the point at infinity).
- 3 – The order n of the point P .

The Elliptic Curve domain parameter can be verified to meet the following requirements [12] and [11]. In order to avoid the Pollard-rho [16] and Pohling-Hellman algorithms for the elliptic curve discrete logarithm problem, it is necessary that the number of F_p -rational points on E , denoted by $\#E(F_p)$, be divisible by a sufficiently large prime n . To avoid the reduction algorithms of Menezes, Okamoto and Vanstone [1] and Frey and Ruck [5], the curve should be non-supersingular (i.e., p should not divide $(p + 1 - \#E(F_p))$). To avoid the attack of Semaev [18] on F_p -anomalous curves, the curve should not be F_p -anomalous (i.e., $\#E(F_p) \neq p$).

The table below (Table 1) shows the ECC mathematical parameters that are used in our proposed protocol.

Table 1. EC. Mathematical Notations

--

Index	Explanation
$(h())$	one-way hash function
p, q	large prime numbers, where $p = 2 \cdot q + 1$
P, Q	Random points over elliptic curve
a, b	Random generated private keys
A, B	Agreed public key
E	non-supersingular elliptic curve
B	$B \in E(F_q)$ with order q
$x(Q)$	x coordinate of point Q
x_u	user's secret key
Y_u	u 's public key with $Y_u = (x_u B) \bmod p$
pw	The user's password

In the following, we will give an introduction to the EC-discrete logarithm problem, to Diffie-Hellman key exchange based on EC and to the elliptic curve based ElGamal signature scheme (ECEGS), which are the core theories used in our proposed protocol.

Let E be an elliptic curve defined over a finite field F_p and let $P \in E(F_p)$ be a point of order n . Given Q where $Q \in E(F_q)$, the elliptic curve discrete logarithm problem is to find the integer $l, 0 \leq l \leq n - 1$, such that $Q = l \cdot P$.

The Diffie-Hellman key agreement protocol runs as follows: Alice selects a random number n_a and computes $Y_a = n_a B$, he sends Y_a to Bob. Similarly Bob computes $Y_b = n_b B$ and sends Y_b to Alice. Alice and Bob generate the same key $K = n_a Y_b B = n_b Y_a = n_a n_b B$.

The ECEGS runs as follows: Alice selects a random number x_a , where $2 \leq x_a \leq n - 2$, as his secret key and computes the corresponding public key $Y_a = x_a B$. Therefore the public key and the private key are (E, Y_a, B, n) and x_a . To generate a signature for a message m , Alice will select a random number k , where $2 \leq k \leq n - 2$ computes $R = kB$ and computes $r = x(KB) \bmod n$. If $r \neq 0$, then computes $s = K^{-1}(h(m) + x_a r) \bmod n$. The couple (R, s) will be Alice's signature of m . To verify the signature Bob will first confirm that r and $s \in [2, n - 2]$ and then computes $v_1 = sR$ and $v_2 = h(m)B + rY_a$. Finally Bob will accept the signature if and only if $(v_1 == v_2)$.

3.2 EC-SAKA Protocol Description

The main goal of our proposed protocol is to achieve mutual authentication and key confirmation between the client and the server in order to establish a secure channel. Our proposed applies the ECEGS to the SKA protocol to enhance the safely level and protocol simplification in terms of computational and communications load. Our proposed protocol consists of three flows and it is illustrated in [Figure 1](#).

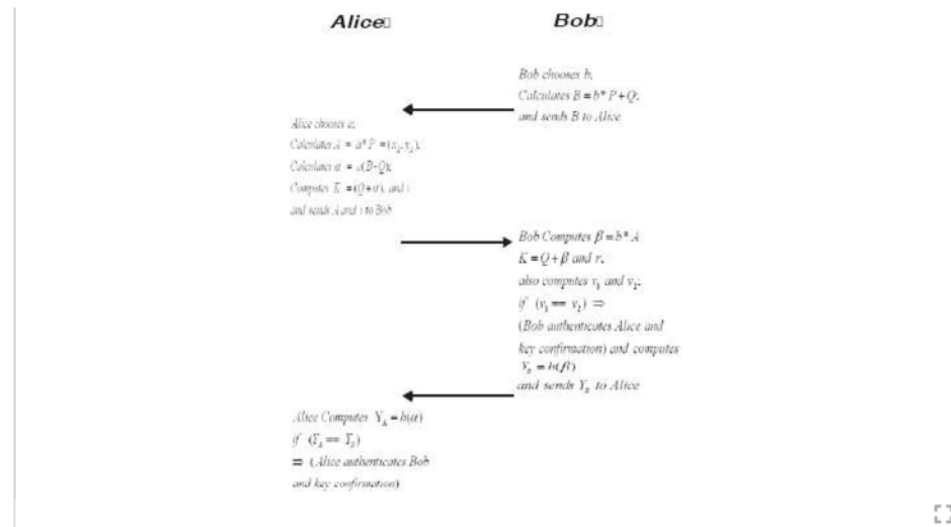


Figure 1 The EC-SAKA Protocol Scheme

Before running the authentication procedure, the client select an elliptic curve $E(Z_p)$ defined on Z_p . Alice, select a random point P over the elliptic curve with order n . n is a large prime number. Alice chooses a password pw , computes $x = h(pw)$ and calculates Q where $Q = x * P$. In addition, Alice generates strong number p and q where $p = 2 * q + 1$. Once the following parameters (E, Q, P, p, q, pw) are generated, Alice transfers (E, Q, P, n) to the server, Bob, in a secure way. Once these steps are done, the session key generation procedure will be executed as follow:

Within the first flow, Bob chooses a random challenge b , where $1 \leq b \leq n - 1$, then he calculates the point B where

$$B = b * P + Q \quad 1$$

[View Source](#)

Finally he sends B to Alice.

Within the second flow, Alice chooses a random challenge a , where $1 \leq a \leq n - 1$, then computes A where

$$A = a * P = (x_A, y_A) \quad 2$$

[View Source](#)

and calculates α where

$$\alpha = a(B - Q) \quad 3$$

[View Source](#)

and $K = Q + \alpha$. In addition, Alice calculates $r = (x_A) \bmod(n)$ and checks that $r \neq 0$. If so, then Alice computes

$$i = a^{-1}(h(\alpha) + x * r) \bmod(n) \quad 4$$

[View Source](#)

Finally (A, i) becomes the signatures pair and Alice transfers A and i to the server.

Within the third flow, Bob computes

$$\beta = h * A \quad 5$$

[View Source](#)

Computes $K = Q + \beta$ and computes $r = x_A \bmod n$. To verify the signature, first Bob checks that $(r, i) \in [2, n - 2]$. If so, then Bob computes

$$v_1 = i * A \quad 6$$

[View Source](#)

and calculates

$$v_2 = (h(\beta)P) + r * Q \quad 7$$

[View Source](#)

Finally, Bob checks if $(v_1 == v_2)$, if so, Bob authenticates Alice and Bob can be confirmed that Alice has actually established the same shared session key. Then Bob computes:

$$Y_B = h(\beta) \quad 8$$

[View Source](#)

and finally he sends Y_B to Alice.

In order to authenticate Bob, Alice will compute:

$$Y_A = h(\alpha) \quad 9$$

[View Source](#)

and then Alice will verify the value of Y_A by checking that $(Y_A == Y_B)$, if so, if they match, then Alice authenticates Bob and Alice can be confirmed that Bob has actually established the same shared session key with her.

Finally, Alice and Bob agree on the common session key K_s where

$$K_s = h(ID(Alice) || ID(Bob) || K) \quad 10$$

[View Source](#)

Both sides will agree on the session Key K_s if all steps are executed correctly. Once the protocol run completes successfully, both parties may use K_s to encrypt subsequent session traffic in order to create a confidential communication channel.

SECTION 4

Security Analysis

In the following section, we will analyze the security of our proposed protocol.

The EC-SAKA protocol is considered to be a secure authenticated key establishment protocol, if it satisfies the following properties:

- **Passive attack:** Suppose that Oscar the attacker perform a passive attack, then the session will terminate with both parties accepting. That is, Bob and Alice successfully identify themselves to each other, and they both compute the session key. So, Oscar, the adversary, cannot compute any information about the common shared session key K_s by assuming the intractability of the elliptic curve discrete algorithms problem. Therefore the EC-SAKA protocol resists against the passive attack.
- **Man in the middle attack (or active attack):** Suppose that an attacker, Oscar, intercepts B and replaces it with B' , Oscar then receives A and i from Alice. He would like to replace i by i' , as before. However, this means that he must calculate α where $\alpha = a(B' - Q)$ but unfortunately for Oscar, he can not compute the value of α because he does not know the value of Q neither the value of a . So, Oscar will not be able to compute K , neither K_s . Therefore the EC-SAKA protocol thwarts the man in-the-middle attack.
- **Dictionary attack:** In dictionary attack, the attacker finds the real password by repeating a process of guessing the password of legal client and applying the passwords. The dictionary could be performed in offline or online mode. In our proposed protocol, it is impossible to get the real password since a one way hash function is applied to the password and during the protocol process, the shared Key K used in the calculation of K_s is calculated from b and a which are generated every new session; and by assuming the intractability of elliptic curve discrete logarithm problem Therefore the EC-SAKA protocol thwarts the offline and online dictionary attack.
- **Known-key attack:** In this attack, an adversary will capture the session key from an eavesdropped session. In our proposed protocol, the client and the server both generates new b and a every new session, and in addition the shared key K is generated with every new session also. Thus our proposed protocol is secure against known key attacks assuming that the elliptic discrete problem is intractable.
- **Perfect forward secrecy:** The perfect forward secrecy is that an exposed password does not enable an attacker to derive session keys of past communication sessions. In our protocol, the security of perfect forward secrecy is based upon the assumption that the elliptic curve discrete problem is intractable and on the value of the key K . Even if the attacker knew the correct password, the attacker still cannot compute the previous session keys because K_s is derived from the shared key K which is generated from the value of a and b . Therefore, the EC-SAKA protocol satisfies the property of perfect forward secrecy.
- **Resilience to server compromised:** if the host's password file is compromised, an adversary can not use it to impersonate legitimate user since the server does not store the file of the password, instead the value of Q where $Q = x * P$ is stored. Thus, Oscar has to solve the elliptic curve discrete logarithm problem in order to retrieve the value of password. The EC-SAKA protocol provides resilience to server compromise.

SECTION 5

Performance Evaluation: Efficiency and Comparison

Computation cost and communication cost are the most important aspects of password authentication protocols which affect the overall performance. They include number of steps, exponentiations, large blocks, symmetric encryption and decryption, hash functions and random numbers. In this section, we compare the

EC-SAKA protocol with the following protocols: Leakage-Resilient Authenticated Key Exchange (LR-AKE) protocol, Simple Key Agreement (SKA) protocol, secure Remote Password (SRP) protocol, EC-SRP, Simple Password Exponential Key Exchange (B-SPEKE) protocol, Password-Authenticated Key Exchange (PAK-X and PAK-RY) protocols and Authentication Memorable Password (AMP) protocol. The comparison is done in terms of number of steps, random numbers, exponentiations and hash functions. Table 2 shows the compared result for number of steps and exponentiation. Table 3 shows the compared result for random numbers and hash functions numbers.

It is clear from Table 2 that the EC-SAKA protocol has the minimal cost in terms of number of steps and exponentiations compared with these above protocols. We can easily notice that B-SPEKE [4], SRP [20], AKEECC [7] and AMP [10] require 4 rounds while PAK-RY [3], PAK-X [13], SKA [17], LR-AKE [6] and EC-SAKA require 3 rounds. In addition, the computational load was clearly improved using EC-SAKA protocol because, as noted in table 2, EC-SAKA requires 1 exponentiations, one for the client and nothing for the server. While for the other protocols, including SKA, LR-AKE, AKEECC [7] and EC-SRP [7], they require at least 4 exponentiations.

Table 2. Comparison of Performance-1-

Protocol	Rounds	Exponentiations		
		Client	Server	Total
B-SPEKE	4	3	4	7
SRP	4	3	3	6
AMP	4	2	3	5
PAK-RY	3	5	4	9
PAK-X	3	5	4	9
SKA	3	2	3	5
LR-AKE	3	3	2	5
AKEECC	4	2	2	4
EC-SRP	3	2	2	4
EC-SAKA	3	1	0	1

Table 3. Comparison of Performance-2-

Protocol	Random N.	Hash Function N.
SRP	2	6
AMP	2	9
PAK-RY	3	8
PAK-X	3	10
SKA	2	7
LR-AKE	2/4	6
AKEECC	2	6
EC-SRP	3	5
EC-SAKA	2	5

From Table 3, we can easily notice that the EC-SAKA protocol requires 2 random numbers and 5 hash functions while all the other protocols require more. In addition, for the EC-SRP and AKEECC protocols described in [7], we can easily notice that our protocol is better than these two protocols in terms of hash functions numbers. For the EC-SRP protocols described in [7], EC-SRP protocol was proposed for a one way authentication while our proposed protocol, EC-SAKA, provides mutual authentication.

SECTION 6

Conclusion

Wireless network access are nowadays very important for users. This paper describes a new network access mechanism for wireless local area networks. In this paper, we introduce a fast and secure authenticated key agreement protocol based on elliptic curve cryptography and provides mutual authentication and explicit key establishment. We also give a formal approach to prove its security based on the previous work by [7]. Our scheme is simple, easy to realize, and secure against both passive and active attacks. It also resists many others attacks as described in [section 4](#). Our proposed protocol is compared to well-known protocols such as B-SPEKE, SRP, EC-SRP, AKEECC, PAK-RY, PAK-X, AMP, SKA and LR-AKE in terms of communication and computation cost and the results were well discussed in the previous section.

ACKNOWLEDGMENTS

The authors would like to thanks the following departments RST-INT Evry and FOE-UL for their support and comments. Their suggestions and observations were extremely helpful throughout this paper.

[Authors](#)[Figures](#)[References](#)[Citations](#)[Keywords](#)[Metrics](#)[Back to Results](#)[IEEE Personal Account](#)[CHANGE](#)[USERNAME/PASSWORD](#)[Purchase Details](#)[PAYMENT OPTIONS](#)[VIEW PURCHASED
DOCUMENTS](#)[Profile Information](#)[COMMUNICATIONS
PREFERENCES](#)[PROFESSION AND
EDUCATION](#)[TECHNICAL INTERESTS](#)[Need Help?](#)[US & CANADA: +1 800
678 4333](#)[WORLDWIDE: +1 732
981 0060](#)[CONTACT & SUPPORT](#)[Follow](#)

[About IEEE Xplore](#) | [Contact Us](#) | [Help](#) | [Accessibility](#) | [Terms of Use](#) | [Nondiscrimination Policy](#) | [IEEE Ethics Reporting](#) | [Sitemap](#) | [IEEE Privacy Policy](#)

A public charity, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

© Copyright 2025 IEEE - All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies.