

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD.;
SAMSUNG ELECTRONICS AMERICA, INC.,
Petitioner

v.

NETWORK-1 TECHNOLOGIES, INC.,
Patent Owner.

Case No. IPR2025-IPR2026-00119
U.S. Patent No. 11,916,893

**DECLARATION OF PAUL S. MIN, PH.D. IN SUPPORT OF
PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 11,916,893**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. BACKGROUND AND QUALIFICATIONS	1
III. INFORMATION CONSIDERED	6
IV. RELEVANT LEGAL STANDARDS	6
A. Claim Interpretation	6
B. Perspective of One of Ordinary Skill in the Art.....	7
C. Obviousness.....	7
V. LEVEL OF ORDINARY SKILL IN THE ART	10
VI. SUMMARY OF MY OPINIONS	11
VII. TECHNOLOGY BACKGROUND.....	12
A. Secure Communication	12
1. Confidentiality	12
a. Encryption Methods	13
b. Key Establishment	15
c. Cryptographic Standards/Requirements	21
2. Authentication	25
3. Integrity and Replay Protection	26
B. GlobalPlatform	29
1. Secure Elements and Security Domains	30
2. Remote Management and Secure Channels.....	31
C. SIMs and eSIMs/eUICCs	33
VIII. THE '893 PATENT AND ITS FILE HISTORY	38
IX. PRIORITY DATE	44
X. CLAIM CONSTRUCTION	44
XI. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART	45
A. Park (Ex-1005)	45
B. GlobalPlatform (Ex-1006).....	46

TABLE OF CONTENTS
(continued)

	Page
C. AbiChar (Ex-1007).....	50
D. X9.63-Overview (Ex-1008)	50
E. Haggerty (Ex-1012).....	52
F. Pierce (Ex-1013).....	53
G. Konstantinou (Ex-1014).....	54
H. Nix175 (Ex-1016).....	55
XII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS	55
A. Ground 1: Claims 1, 3, 6-11, 13-17 are obvious over Park in view of GlobalPlatform and AbiChar	55
1. A POSITA would have been motivated to combine Park’s teachings with GlobalPlatform’s and AbiChar’s teachings and would have had a reasonable expectation of success.....	55
a. A POSITA would have turned to GlobalPlatform for implementation-level details of Park’s security requirements.	55
b. A POSITA would have understood that Abi-Char’s elliptic curve Diffie-Hellman key agreement could be implemented within Park’s SPA to provide an efficient asymmetric key-exchange mechanism.....	60
2. Independent Claim 1	63
a. Element 1[pre]: A mobile device for communicating with a wireless network, the mobile device comprising:.....	63
b. Element 1[a]: a first memory configured to store an embedded universal integrated circuit card (eUICC) identity;	65

TABLE OF CONTENTS

(continued)

	Page
c. Element 1[b]: a random number generator operably connected to a processor connected to a second memory configured to generate a random number for an eUICC private key corresponding to an eUICC public key;	68
d. Element 1[c]: a radio including one or more transmit antennas and one or more receiving antennas configured to:.....	71
e. Element 1[c][1]: a. transmit, to a subscription manager, the eUICC identity and the eUICC public key; and.....	72
f. Element 1[c][2]: b. receive, from the subscription manager, i) an eUICC profile comprising network parameters, a key K, and a subscriber identity and ii) a symmetric key; and	74
g. Element 1[d]: an eUICC associated with the eUICC identity and configured to	78
h. Element 1[d][1]: a. derive a profile key using an elliptic curve Diffie-Hellman (ECDH) key exchange with the eUICC private key and a subscription manager public key;	79
i. Element 1[d][2]: b. decrypt a first portion of the eUICC profile using the profile key;	80
j. Element 1[d][3]: c. receive the symmetric key from a network application operating in the mobile device;	81
k. Element 1[d][4]: d. decrypt a second portion of the eUICC profile using the symmetric key, the second portion comprising the key K and the subscriber identity, wherein the first portion and the second portion are distinct; and	82

TABLE OF CONTENTS
(continued)

	Page
1. Element 1[d][5]: e. generate a response value for authentication of the mobile device with the wireless network using the key K.....	84
3. Dependent Claims 3, 6-11, 13-17	86
a. Claim 3: The mobile device of claim 1, wherein the eUICC operates within a universal integrated circuit card within the mobile device.	86
b. Claim 6: The mobile device of claim 1, wherein the network application is configured to communicate with i) the wireless network and ii) the eUICC in the mobile device using a system bus.....	86
c. Claim 7: The mobile device of claim 1, wherein the eUICC comprises computer executable instructions for a processor in the mobile device, and wherein the computer executable instructions are stored within the memory.	88
d. Claim 8: The mobile device of claim 1, wherein the eUICC comprises a package soldered onto a circuit board of the mobile device.	88
e. Claim 9: The mobile device of claim 1, further comprising a user interface configured to receive user identification information before the mobile device receives the symmetric key.	88
f. Claim 10: The mobile device of claim 1, wherein the first portion of the eUICC profile includes the network parameters, the network parameters comprising a list of numbers associated with a mobile network operator.....	89
g. Claim 11: The mobile device of claim 1, wherein the network parameters comprise a mobile country code (MCC) and a mobile network code (MNC) associated with the wireless network.	90

TABLE OF CONTENTS
(continued)

	Page
h. Claim 13: The mobile device of claim 1, wherein the subscriber identity comprises an International Mobile Subscriber Identity (IMSI).....	90
i. Claim 14: The mobile device of claim 1, wherein the radio is further configured to receive, from the subscription manager, a ciphertext for the symmetric key, wherein the mobile device is configured to decrypt the ciphertext using at least the eUICC private key.	91
j. Claim 15: The mobile device of claim 1, wherein the mobile device further comprises at least one of a wireless handset, a cellular phone, a smartphone, a tablet computer, a laptop, a tracking device, and a circuit board with the radio.....	91
k. Claim 16: The mobile device of claim 1, wherein the second memory is the same memory as the first memory.....	92
l. Claim 17: The mobile device of claim 1, wherein the one or more receiving antennas are the one or more transmit antennas.....	93
B. Ground 2: Claims 1, 3, 6-11, 13-17 are obvious over Park in view of GlobalPlatform and X9.63-Overview.....	93
1. A POSITA would have been motivated to combine Park's teachings with GlobalPlatform's and X9.63-Overview's teachings and would have had a reasonable expectation of success.....	93
2. Independent Claim 1	95
a. Elements 1[pre]-1[a], 1[c], 1[c][2]-1[d], 1[d][2]-1[d][5]:.....	95

TABLE OF CONTENTS

(continued)

	Page
<ul style="list-style-type: none"> b. Element 1[b]: a random number generator operably connected to a processor connected to a second memory configured to generate a random number for an eUICC private key corresponding to an eUICC public key; c. Element 1[c][1]: a. transmit, to a subscription manager, the eUICC identity and the eUICC public key; and..... d. Element 1[d][1]: a. derive a profile key using an elliptic curve Diffie-Hellman (ECDH) key exchange with the eUICC private key and a subscription manager public key;..... 	95 97 98
3. Dependent Claims 3, 6-11, 13-17	99
C. Ground 3: Claims 2 and 12 are obvious over Grounds 1 or 2 in view of Haggerty	99
<ul style="list-style-type: none"> 1. A POSITA would have been motivated to add Haggerty to Park-GlobalPlatform-AbiChar or Park-GlobalPlatform-X9.63-Overview and would have had a reasonable expectation of success..... 2. Claim 2: The mobile device of claim 1, wherein the eUICC is further configured to receive a profile identity for the eUICC profile as a plaintext. 3. Claim 12: The mobile device of claim 1, wherein the radio is further configured to receive, from the subscription manager, the eUICC profile using transport layer security. 	99 103 103
D. Ground 4: Claims 4-5 are obvious over Grounds 1 or 2 in view of Pierce.....	104
<ul style="list-style-type: none"> 1. A POSITA would have been motivated to add Pierce's to Park-GlobalPlatform-AbiChar or Park-GlobalPlatform-X9.63-Overview and would have had a reasonable expectation of success..... 2. Claim 4: The mobile device of claim 1, wherein the 	104

TABLE OF CONTENTS

(continued)

	Page
random number generator is configured to generate the random number in response to input from at least one of a clock and a sensor.	105
Claim 5: The mobile device of claim 1, wherein the random number generator is configured to generate the random number from a seed value, wherein the seed value comprises data from at least one of a sensor, the radio, a bus, a clock, a physical interface, the memory, and an operating system.	105
E. Ground 5: Claims 10-11 are obvious over Grounds 1 or 2 in view of Konstantinou	105
1. A POSITA would have been motivated to add Konstantinou to Park-GlobalPlatform-AbiChar or Park-GlobalPlatform-X9.63-Overview and would have had a reasonable expectation of success.....	105
2. Claim 10: The mobile device of claim 1, wherein the first portion of the eUICC profile includes the network parameters, the network parameters comprising a list of numbers associated with a mobile network operator.....	106
Claim 11: The mobile device of claim 1, wherein the network parameters comprise a mobile country code (MCC) and a mobile network code (MNC) associated with the wireless network.....	106
F. Ground 6: Claims 1-17 are obvious over Nix175 in view of Park and GlobalPlatform	107
1. A POSITA would have been motivated to combine Nix175, Park, and GlobalPlatform and would have had a reasonable expectation of success.....	107
2. Independent Claim 1	113
a. 1[pre]: A mobile device for communicating with a wireless network, the mobile device comprising:	113

TABLE OF CONTENTS

(continued)

	Page
b. 1[a]: a first memory configured to store an embedded universal integrated circuit card (eUICC) identity;	114
c. 1[b]: a random number generator operably connected to a processor connected to a second memory configured to generate a random number for an eUICC private key corresponding to an eUICC public key;	114
d. 1[c]: a radio including one or more transmit antennas and one or more receiving antennas configured to:.....	116
e. 1[c][1]: a. transmit, to a subscription manager, the eUICC identity and the eUICC public key; and.....	116
f. 1[c][2]: b. receive, from the subscription manager, i) an eUICC profile comprising network parameters, a key K, and a subscriber identity and ii) a symmetric key; and	117
g. 1[d]: an eUICC associated with the eUICC identity and configured to	121
h. 1[d][1]: a. derive a profile key using an elliptic curve Diffie-Hellman (ECDH) key exchange with the eUICC private key and a subscription manager public key;	121
i. 1[d][2]: b. decrypt a first portion of the eUICC profile using the profile key;	122
j. 1[d][3]: c. receive the symmetric key from a network application operating in the mobile device;.....	123
k. 1[d][4]: d. decrypt a second portion of the eUICC profile using the symmetric key, the second portion comprising the key K and the subscriber identity, wherein the first portion and the second portion are distinct; and	124

TABLE OF CONTENTS
(continued)

	Page
1. 1[d][5]: e. generate a response value for authentication of the mobile device with the wireless network using the key K.....	126
3. Dependent Claims 2-17.....	127
a. Claim 2: The mobile device of claim 1, wherein the eUICC is further configured to receive a profile identity for the eUICC profile as a plaintext.	127
b. Claim 3: The mobile device of claim 1, wherein the eUICC operates within a universal integrated circuit card within the mobile device.	128
c. Claim 4: The mobile device of claim 1, wherein the random number generator is configured to generate the random number in response to input from at least one of a clock and a sensor.....	129
Claim 5: The mobile device of claim 1, wherein the random number generator is configured to generate the random number from a seed value, wherein the seed value comprises data from at least one of a sensor, the radio, a bus, a clock, a physical interface, the memory, and an operating system.....	129
d. Claim 6: The mobile device of claim 1, wherein the network application is configured to communicate with i) the wireless network and ii) the eUICC in the mobile device using a system bus.....	129
e. Claim 7: The mobile device of claim 1, wherein the eUICC comprises computer executable instructions for a processor in the mobile device, and wherein the computer executable instructions are stored within the memory.	130
f. Claim 8: The mobile device of claim 1, wherein the eUICC comprises a package soldered onto a circuit board of the mobile device.	130

TABLE OF CONTENTS
(continued)

	Page
g. Claim 9: The mobile device of claim 1, further comprising a user interface configured to receive user identification information before the mobile device receives the symmetric key.	130
h. Claim 10: The mobile device of claim 1, wherein the first portion of the eUICC profile includes the network parameters, the network parameters comprising a list of numbers associated with a mobile network operator. Claim 11: The mobile device of claim 1, wherein the network parameters comprise a mobile country code (MCC) and a mobile network code (MNC) associated with the wireless network.	131
i. Claim 12: The mobile device of claim 1, wherein the radio is further configured to receive, from the subscription manager, the eUICC profile using transport layer security.	132
j. Claim 13: The mobile device of claim 1, wherein the subscriber identity comprises an International Mobile Subscriber Identity (IMSI).	132
k. Claim 14: The mobile device of claim 1, wherein the radio is further configured to receive, from the subscription manager, a ciphertext for the symmetric key, wherein the mobile device is configured to decrypt the ciphertext using at least the eUICC private key.	132
l. Claim 15: The mobile device of claim 1, wherein the mobile device further comprises at least one of a wireless handset, a cellular phone, a smartphone, a tablet computer, a laptop, a tracking device, and a circuit board with the radio.	133
m. Claim 16: The mobile device of claim 1, wherein the second memory is the same memory as the first memory.	133

TABLE OF CONTENTS
(continued)

	Page
n. Claim 17: The mobile device of claim 1, wherein the one or more receiving antennas are the one or more transmit antennas.....	134
XIII. CONCLUSION.....	134

LIST OF EXHIBITS¹

Ex. No.	Description
Ex-1001	U.S. Patent No. 11,916,893 (“the ’893 Patent”)
Ex-1002	Declaration of Dr. Paul S. Min
Ex-1003	Curriculum Vitae of Dr. Paul S. Min
Ex-1004	Prosecution History of U.S. Patent No. 11,916,893
Ex-1005	Jaemin Park et al., Secure Profile Provisioning Architecture for Embedded UICC, Int’l Conference on Availability, Reliability & Security 297 (2013) (“Park”)
Ex-1006	GlobalPlatform Card Specification, Version 2.2.1, Public Release (Jan. 2011) (“GlobalPlatform”)
Ex-1007	Pierre E. Abi-Char et al., A Fast and Secure Elliptic Curve Based Authenticated Key Agreement Protocol For Low Power Mobile Communications, Int’l Conference on Next Generation Mobile Applications, Services and Technologies (2007) (“AbiChar”)
Ex-1008	ANSI X9.63 Overview, Key Agreement and Key Transport Using Elliptic Curve Cryptography, Simon Blake-Wilson, Certicom (2000) (“X9.63-Overview”)
Ex-1009	INTENTIONALLY LEFT BLANK
Ex-1010	INTENTIONALLY LEFT BLANK
Ex-1011	GlobalPlatform Card Secure Channel Protocol ’11’ Card Specification v2.2 – Amendment F, Version 1.0 (May 2015) (“SCP11”)

¹ Four-digit pin citations that begin with 0 are to the branded numbers added by Samsung in the bottom right corner of the exhibits. All other pin citations are to original page, column, paragraph, or line numbers.

Ex. No.	Description
Ex-1012	U.S. Patent Pub. No. 2013/0227646 to Haggerty (“Haggerty”)
Ex-1013	U.S. Patent Pub. No. 2009/0323967 to Pierce (“Pierce”)
Ex-1014	U.S. Patent Pub. No. 2010/0267383 to Konstantinou (“Konstantinou”)
Ex-1015	Declaration of Simon Blake-Wilson, author of ANSI X9.63-Overview
Ex-1016	U.S. Patent No. 9,100,175 (“Nix175”)
Ex-1017	INTENTIONALLY LEFT BLANK
Ex-1018	Declaration of Tono Aspinall of GlobalPlatform, Inc.
Ex-1019	Declaration of Gordan MacPherson of IEEE
Ex-1020	Claim Mapping Table
Ex-1021	Colin Boyd and Anish Mathuria, “Protocols for Authentication and Key Establishment” (Springer, 2003) (“Boyd & Mathuria”)
Ex-1022	Whitfield Diffie & Martin E. Hellman, <i>New Directions in Cryptography</i> , 22 IEEE Trans. Info. Theory 644 (1976)
Ex-1023	NIST, SP 800-57, Recommendation for Key Management: Part 1 — General (Rev. 3) (July 2012)
Ex-1024	NIST, SP 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping (Dec. 2012)
Ex-1025	NIST, SP 800-56A, Revision 2, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (May 2013)
Ex-1026	GlobalPlatform Card Technology, Secure Channel Protocol 03, Card Specification v 2.2 – Amendment D, Version 1.0 (Apr. 2009)

Ex. No.	Description
Ex-1027	Reprogrammable SIMs: Technology, Evolution and Implications, Final Report, Ofcom (Sept. 25, 2012)
Ex-1028	ETSI TS 103 383 v12.1.0, “Smart Cards; Embedded UICC; Requirements Specification (Release 12),” (June 2013)
Ex-1029	Network-1’s Infringement Contentions for the ’893 Patent, filed on September 30, 2025, as Exhibit C to Network-1 Technologies, Inc.’s P.R. 3-1 Disclosure of Asserted Claims and Infringement Contentions

I. INTRODUCTION

1. I, Paul S. Min, Ph.D., submit this declaration to state my opinions on the matter described below.

2. I have been retained by Petitioners Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc. (collectively, “Petitioner”), as an independent expert in this proceeding before the United States Patent and Trademark Office. I am being compensated at my usual and customary rate of \$500 per hour. No part of my compensation depends on the outcome of this proceeding, and I have no other interest in this proceeding.

3. I understand that this proceeding involves U.S. Patent No. 11,916,893 (“the ’893 Patent”) (Ex-1001), which I understand is currently assigned to Network-1 Technologies, Inc. (“PO”). I have been asked to consider the validity of Claims 1-17 of the ’893 Patent. My opinions and the bases for my opinions are set forth below.

II. BACKGROUND AND QUALIFICATIONS

4. I believe that I am well qualified to serve as a technical expert in this matter based upon my educational and work experience. My curriculum vitae (“CV”) is submitted as Exhibit 1003.

5. I received a B.S. degree in Electrical Engineering in 1982, an M.S. degree in Electrical Engineering in 1984, and a Ph.D. degree in Electrical Engineering in 1987 from the University of Michigan in Ann Arbor. I received

several academic honors, including my B.S. degree with honors, a best graduate student award and a best teaching assistant award during my M.S. study, and a best paper award from a major international conference for reporting results from my Ph.D. thesis.

6. After receiving my Ph.D., I worked at Bellcore in New Jersey from August 1987 until August 1990. At Bellcore, I was responsible for evolving the public switched telephone network (POTS) into a multi-services voice and data network that incorporated packet switches, optical technologies, and wireless technologies.

7. In September 1990, I joined the faculty at Washington University in St. Louis. In July 1996, I was promoted to an Associate Professor of Electrical Engineering with tenure. I am currently a Senior Professor at Washington University of Electrical and Systems Engineering. I have also served as the Chair of the Graduate Curriculum (2000-2002) and the Chair of the Undergraduate Curriculum (2011-2014) for the Electrical and Systems Engineering department.

8. At Washington University, I have conducted research in communication, computing, and related electronic hardware and software. My research group has pioneered a new paradigm for designing electronic circuits that can alleviate the speed and performance mismatch against optical technology. I have

received several grants from the U.S. Federal Agencies, including the National Science Foundation and the Defense Advanced Research Project Agency, and numerous contracts from companies and organizations around the world. Specifically related to the technology matters in this Investigation, I have researched a variety of wireless communication technologies, including IEEE 802.11, IEEE 802.15, 3GPP CDMA, 3GPP WCDMA, 3GPP OFDM, 3GPP FDD, 3GPP C-FDMA, and 3GPP TDD. In utilizing these technologies, I have acquired extensive personal knowledge in authentication and verification methods, which include various types of encryptions.

9. As a faculty member at Washington University, I have taught a number of courses in electronics, communication, and computing at both the undergraduate and graduate levels. For example, I have taught communication theory (Washington University ESE 471), transmission and multiplexing (Washington University ESE 571), and signaling and control of communication networks (Washington University ESE 572).

10. I have supervised numerous graduate students, many of whom received a doctoral degree under my guidance. A number of doctoral theses that I have supervised relate specifically to the authentication and encryption in wireless communication technology. In particular, my students and I have published a

number of peer-reviewed articles on access control, resource allocation, scheduling, modulation, mobility management, and multiplexing. Several of these articles received accolades in the field. For example, in 2011, we received a best paper award in 3G WCDMA-related mobility and resource management at the prestigious Mobility 2011 international conference.

11. In addition to my responsibilities as a university faculty member, I have founded two companies. In May 1997, I founded MinMax Technologies, Inc., a fabless semiconductor company that developed switch fabric integrated circuit chips for the Internet. In March 1999, I founded Erlang Technology, Inc., a fabless semiconductor company that focused on the design and development of integrated circuit chips and software for the Internet. One of Erlang's products received a best product of the year award in 2004 from a major trade journal for the electronics industry.

12. Outside my own start-up companies, I have also served various technology and business advisor roles for other companies and organizations around the world. I was the main technical author for one of two winning proposals to the Korean government for CDMA wireless service licenses (1996). I was responsible for designing a commercial scale IS-95 CDMA cellular network, which I understand to be one of the earliest such networks deployed in the world. I worked with

numerous engineers and scientists around the world to implement this commercial scale cellular network before IS-95 CDMA was widely accepted. This provided me with extensive insight into various components of CDMA technology, which by and large are used in WCDMA network. I have also been involved in a semiconductor company that specializes in semiconductor memories such as flash EEPROMs as a board member and as a technical advisor (2007-2011).

13. I am a named inventor on eleven U.S. patents, many of which are directly related to resource allocation, packet processing, and network designing. I have extensively published technical papers in international conferences and journals, technical memoranda and reports, and given a number of seminars and invited talks. Many of these papers are specifically within the context of the wireless communication standards. I have organized several international conferences and served as an international journal editor.

14. I am a member of and have been actively involved in a number of professional organizations. For example, I have served as the Chair of the Saint Louis Section of the IEEE with more than 3,000 members (2014), and a member of the Eta Kappa Nu Honor Society for electrical engineers. I have also been an Ambassador of the McDonnell International Scholars Academy (2007-2013).

III. INFORMATION CONSIDERED

15. In preparing this Declaration, I considered the various materials I discuss in this Declaration, including, for example, the '893 Patent, the references cited by the '893 Patent, the prosecution history of the '893 Patent (including the references cited therein), various background articles and materials referenced in this Declaration, and the prior art references I identify in this Declaration. In addition, my opinions are further based on my education, training, experience, and knowledge in the relevant field.

IV. RELEVANT LEGAL STANDARDS

16. I am not an attorney and offer no legal opinions. For the purposes of my Declaration, I have been informed about certain aspects of the law that are relevant to my analysis, as summarized below.

A. Claim Interpretation

17. I have been informed and understand that in an IPR proceeding, claims are to be interpreted according to the *Phillips* claim construction standard. I understand that under this standard, the meaning of claim terms is considered from the viewpoint of a hypothetical “person of ordinary skill in the art” (“POSITA”) at the time of the alleged invention. I have been informed and understand that claim construction is a matter of law and that the final claim constructions for this proceeding will be determined by the Patent Trial and Appeal Board (“PTAB”).

18. I provide my opinions regarding terms that may require explicit construction to resolve the grounds I present in my Declaration in Section X.

B. Perspective of One of Ordinary Skill in the Art

19. I have been informed and understand that a patent is to be understood from the perspective of a hypothetical POSITA. Such an individual is considered to possess normal skills and knowledge in a particular technical field (as opposed to being a genius). I understand that in considering what the claims of a patent require, what was known prior to that patent, what a prior art reference discloses, and whether an invention is obvious or not, one must use the perspective of such a POSITA.

C. Obviousness

20. I have been informed and understand that a patent claim is obvious under 35 U.S.C. §103, and therefore invalid, if the claimed subject matter, as a whole, would have been obvious to a POSITA as of the priority date of the patent based on one or more prior art references and/or the knowledge of a POSITA.

21. I understand that an obviousness analysis must consider (1) the scope and content of the prior art, (2) the differences between the claims and the prior art, (3) the level of ordinary skill in the pertinent art, and (4) secondary considerations, if any, of non-obviousness (such as unexpected results, commercial success, long-felt but unmet need, failure of others, copying by others, and skepticism of experts).

22. I understand that a prior art reference may be combined with other

references to disclose each element of the invention under 35 U.S.C. §103. I understand that a reference may also be combined with the knowledge of a POSITA, and that this knowledge may be used to combine multiple references. I further understand that a POSITA is presumed to know the relevant prior art. I understand that the obviousness analysis may take into account the inferences and creative steps that a POSITA would employ.

23. In determining whether a prior art reference would have been combined with other prior art or other information known to a POSITA, I understand that the following principles may be considered:

- a. whether the references to be combined involve non-analogous art;
- b. whether the references to be combined are in different fields of endeavor than the alleged invention in the Patent;
- c. whether the references to be combined are reasonably pertinent to the problems to which the inventions of the Patent are directed;
- d. whether the combination is of familiar elements according to known methods that yields predictable results;
- e. whether a combination involves the substitution of one known element for another that yields predictable results;
- f. whether the combination involves the use of a known technique to

improve similar items or methods in the same way that yields predictable results;

- g. whether the combination involves the application of a known technique to a prior art reference that is ready for improvement, to yield predictable results;
- h. whether the combination is “obvious to try”;
- i. whether the combination involves the known work in one field of endeavor prompting variations of it for use in either the same field or a different one based on design incentives or other market forces, where the variations are predictable to a POSITA;
- j. whether there is some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill in the art to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- k. whether the combination requires modifications that render the prior art unsatisfactory for its intended use;
- l. whether the combination requires modifications that change the principle of operation of the reference;
- m. whether the combination is reasonably expected to be a success; and

n. whether the combination possesses the requisite degree of predictability at the time the invention was made.

24. I understand that in determining whether a combination of prior art references renders a claim obvious, it is helpful to consider whether there is some teaching, suggestion, or motivation to combine the references and a reasonable expectation of success in doing so. I understand, however, that a teaching, suggestion, or motivation to combine is not required.

V. LEVEL OF ORDINARY SKILL IN THE ART

25. I am familiar with the level of ordinary skill in the art with respect to the '893 Patent around its filing date and its claimed priority date. Based on my experience working, teaching, and conducting research in the relevant field, and based on my review of the '893 Patent specification, claims, file history, and prior art, I believe one of ordinary skill in the art around the time of the alleged invention of the '893 Patent would have had at least a bachelor's degree in electrical engineering, computer engineering, computer science, or a similar field, and 2–3 years of experience with cellular/WLAN security and mobile devices, with additional education substituting for experience and vice versa.

26. In determining the level of ordinary skill in the art, I considered, for example, the type of problems encountered in the art, prior art solutions to those

problems, the rapidity with which innovations are made, the sophistication of the technology, and the educational level of active workers in the field.

27. I met or exceeded the knowledge and experience that a POSITA would have had in 2013. In 2013, I had a Ph.D. in Electrical Engineering and was a tenured professor in the Department of Electrical and Systems Engineering at Washington University. By then, I had nearly two decades of hands-on experience in wireless communication technology, including the deployment of commercial-scale mobile cellular network. In addition, I had extensive understanding of various cybersecurity methods including encryption, and had built numerous security appliance devices for the Internet.

VI. SUMMARY OF MY OPINIONS

28. As I explain in detail below, it is my opinion that Claims 1-17 of the '893 Patent would have been obvious under 35 U.S.C. § 103 on the following grounds:

- **Ground 1:** Claims 1, 3, 6-11, and 13-17 are obvious over Park (Ex-1005) in view of GlobalPlatform (Ex-1006) and AbiChar (Ex-1007).
- **Ground 2:** Claims 1, 3, 6-11, and 13-17 are obvious over Park in view of GlobalPlatform and X9.63-Overview (Ex-1008).
- **Ground 3:** Claims 2 and 12 are obvious over Grounds 1 or 2 in view of

Haggerty (Ex-1012).

- **Ground 4:** Claims 4-5 are obvious over Grounds 1 or 2 in view of Pierce.
- **Ground 5:** Claims 10-11 are obvious Grounds 1 or 2 in view of Konstantinou (Ex-1014).
- **Ground 6:** Claims 1-17 are obvious over Nix175 (Ex-1016) in view of Park and GlobalPlatform.

VII. TECHNOLOGY BACKGROUND

A. Secure Communication

29. Modern communication systems rely on cryptography to protect the confidentiality, integrity, and authenticity of information exchanged between devices. Whether data is being transmitted over a mobile network, a Wi-Fi connection, or the Internet, cryptographic protocols help ensure that only authorized parties can read or modify the information, and that the communicating entities can verify one another's identities to prevent impersonation, unauthorized access, and man-in-the-middle attacks that could otherwise compromise the confidentiality or integrity of the communication.

1. Confidentiality

30. In a secure communication system, information should only be accessible to authorized parties. *See, e.g.,* Ex-1021, Colin Boyd and Anish Mathuria, "Protocols for Authentication and Key Establishment" (Springer, 2003) (hereinafter

“Boyd & Mathuria”), at 4-5, 14. Encryption is the fundamental mechanism used to maintain confidentiality. Encryption transforms readable data (“plaintext”) into an unintelligible form (“ciphertext”) using a mathematical function and a cryptographic key. Without the appropriate decryption key, the ciphertext cannot feasibly be converted back into plaintext. When the intended recipient receives the encrypted message, they use the corresponding decryption key to decrypt the message and recover the original data in plaintext.

a. Encryption Methods

31. Encryption can be applied at multiple levels within a communication system. It may be used to protect specific pieces of sensitive information—for example, a secret key or credential transmitted between devices—or to secure entire data streams exchanged during an active communication session. In the latter case, both parties typically rely on a session key—a temporary cryptographic key generated for use over the duration of a single session. The session key is used to encrypt and decrypt all messages exchanged during that session, ensuring confidentiality while limiting the potential impact if the key were ever compromised.

32. Encryption schemes that protect either individual data items or session traffic generally fall into two broad categories: symmetric encryption and asymmetric encryption.

(1) Symmetric Encryption

33. Symmetric encryption, also known as shared secret encryption, uses the same cryptographic key for both encryption and decryption. *See, e.g.,* Ex-1023, NIST SP 800-57, at 35.

34. Symmetric encryption offers important advantages—it is computationally efficient, requires relatively few processing resources, and is well-suited for encrypting large volumes of data. For this reason, it is commonly used for encrypting stored data (“data at rest”) or for protecting the contents of an established communication session, using a session key.

35. However, symmetric encryption introduces a key-distribution problem—both parties must already share the same secret key before secure communication can begin. That key must be provisioned securely, either by pre-provisioning (e.g., installation on the device during manufacturing or initialization), secure transmission (e.g., generated by one entity and transmitted securely), or established dynamically through a key-exchange or key-agreement protocol, as described in further detail below.

(2) Asymmetric Encryption

36. Asymmetric encryption, in contrast to symmetric cryptography, uses two different (yet complimentary) keys for encryption and decryption. *See, e.g.,* Ex-1023, NIST SP 800-57, at 36. One key is used to encrypt the message and the other

key is used to decrypt the message. This method of encryption is also known as public key encryption because one of the keys is made public, and distributed freely, while the other key is kept private. A message encrypted with one key can only be decrypted with the other, complimentary key.

37. The main advantage of asymmetric encryption is that the private key never needs to be shared or transmitted. Each participant can generate their own key pair locally, ensuring that the private key remains secret. This eliminates the need for pre-established secure channels. However, asymmetric cryptography is computationally more intensive than symmetric cryptography, making it less suitable for encrypting bulk data. Instead, it is typically used for authentication, digital signatures, and secure key establishment (all described in further detail below), after which a faster symmetric cipher can be used for ongoing data encryption.

b. Key Establishment

38. Not all cryptographic keys can be pre-provisioned—that is, securely installed on devices during manufacturing or initialization. In many modern systems, new keys must be generated, delivered, or updated dynamically to support ongoing operations such as establishing secure communications, rotating credentials, or provisioning profiles. Because these keys (often symmetric keys)

must be shared between the parties without exposing them to interception, cryptographic mechanisms are required for their secure establishment.

39. Broadly speaking, there are two principal approaches to key establishment: key transport and key agreement. *See, e.g.*, Ex-1021, Boyd & Mathuria, at 13.

(1) Key Transport

40. In a key transport mechanism, one entity—such as a server, or one of the parties to the communication—generates a symmetric key and securely transmits it to another entity. *See, e.g.*, Ex-1021, Boyd & Mathuria, at 13. Because the key itself must travel over a network, it is typically encrypted, or “wrapped,” with another trusted key before transmission. *See, e.g.*, Ex-1023, NIST SP 800-57, at 40. The wrapping key may be a pre-shared symmetric key (e.g., a long-term key-encrypting key (“KEK”)), a public key belonging to the recipient, or a session key derived from an earlier authenticated exchange. The encrypted (wrapped) key is then sent to the recipient, who decrypts it using the corresponding private or shared key, thereby recovering the transported key. This technique helps to ensure that even if the transmission is intercepted, the attacker cannot learn the transported key.

41. Key transport is widely used in systems that periodically need to inject new symmetric keys into secure elements or devices. For example, in the

GlobalPlatform Secure Channel Protocols SCP02 and SCP03, discussed below, PUT KEY commands are used to provision new keys to a smart card. The new key data sent using the PUT KEY command is encrypted under a Data Encryption Key (DEK) so that it remains protected in transit.

(2) Key Agreement

42. In contrast to key transport, a key agreement mechanism allows both parties to independently generate the same shared secret key, without transmitting that key. *See, e.g.,* Ex-1021, Boyd & Mathuria, at 137. This eliminates the need for one side to generate and send the key, thereby avoiding any potential for exposure of the key material on the communication channel.

(a) Diffie-Hellman Key Exchange

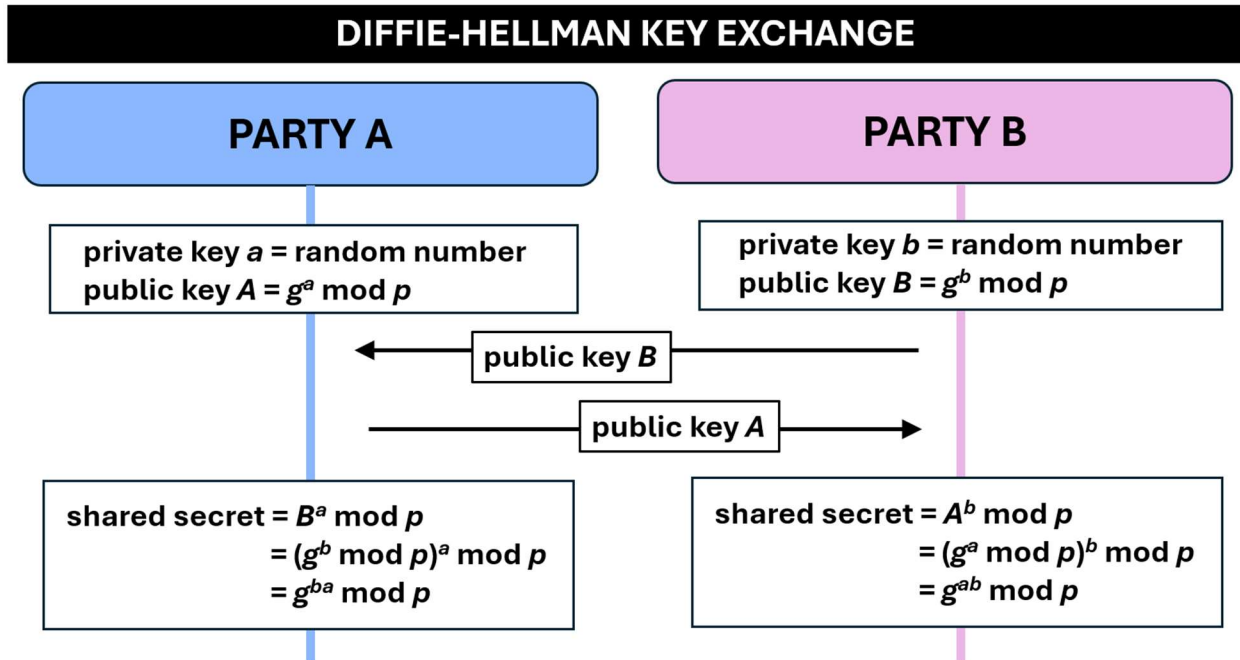
43. The most widely used example is the Diffie-Hellman key exchange algorithm. Indeed, “sometimes key agreement is even used synonymously with the Diffie-Hellman technique.” Ex-1021, Boyd & Mathuria, at 140. The algorithm was first proposed by Whitfield Diffie and Martin Hellman in 1976. *See* Ex-1022, Diffie and Hellman, “New Directions in Cryptography” IEEE Transactions on Information Theory, Vol. 22, No. 6 (November 1976). The Diffie-Hellman algorithm uses public-private key pairs to establish a shared secret between the two parties, which can then be used as a symmetric key to encrypt and decrypt messages. Although the public keys are exchanged openly, the shared secret can only be computed by a party

that knows one of the private keys, making it infeasible for an eavesdropper to derive.

44. The Diffie-Hellman key exchange operates as follows:

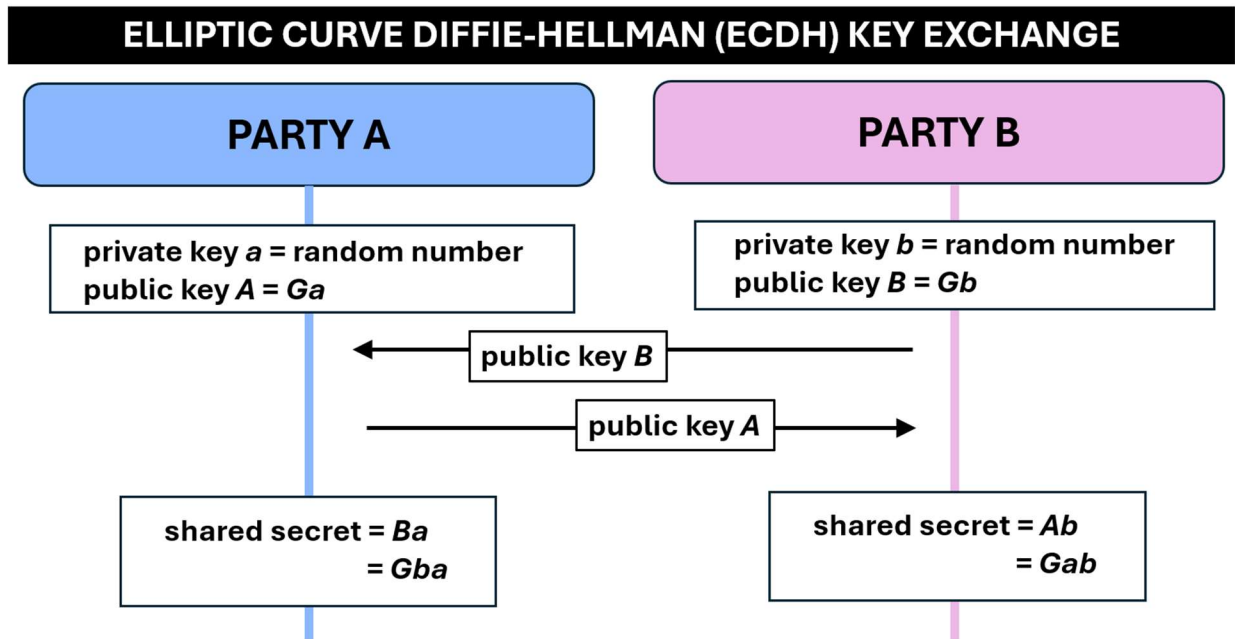
- (1) First, both parties agree on public parameters: a large prime number p and a base g .
- (2) Each party chooses a private random number (a and b , respectively), and then computes a corresponding public key ($A = g^a \bmod p$ and $B = g^b \bmod p$, respectively). Here, the modulus (“mod”) operation means that after computing g^a or g^b , the result is divided by p , and only the remainder is kept. This operation confines all results to the range 0 through $p - 1$.
- (3) The parties then exchange public keys.
- (4) Each party then raises the other party’s public key to its own private exponent, generating $B^a \bmod p = (g^b \bmod p)^a \bmod p$; and $A^b \bmod p = (g^a \bmod p)^b \bmod p$, respectively. As a result of the mathematical properties of exponentials (i.e., $(x^y)^z = x^{yz}$) and the modulus operation (i.e., $(x \bmod y)^z \bmod y = x^z \bmod y$), in this manner, both parties generate the same shared secret, $S = g^{ab} \bmod p$.

See, e.g., Ex-1021, Boyd & Mathuria, at 141-42. These steps are illustrated in the diagram below:



(b) Elliptic Curve Diffie-Hellman

45. The Elliptic Curve Diffie-Hellman Key Exchange (“ECDH”) is a variant of the above-described Diffie-Hellman Key Exchange, in which the algebra of elliptic curves is used instead of modular exponentiation. In ECDH, each party selects a random private key and computes a public key through scalar multiplication on an elliptic curve. Public keys are exchanged, and the shared secret is then derived by multiplying the other party’s public key by one’s own private key, as illustrated below, where G is a point on an elliptic curve:



46. The security of both the classic Diffie-Hellman exchange and the elliptic curve variant relies on the intractability of a discrete logarithm problem. In particular, for the classic Diffie-Hellman algorithm, an attacker must find the secret exponent a , after having intercepted the corresponding public key, $A = g^a \bmod p$. Similarly, for the elliptic curve variant, ECDH, an attacker must find the secret scalar a , after having intercepted the corresponding public key, $A = aG$. While both tasks are believed to be computationally infeasible for properly chosen parameters, the elliptic-curve discrete logarithm problem is much harder per bit of key length than the modular exponentiation problem. Accordingly, the use of elliptic-curve cryptography (ECC) in the ECDH key exchange provides equivalent cryptographic strength using much shorter keys, as compared to the traditional Diffie-Hellman

algorithm. *See, e.g.,* Ex-1023, NIST Special Publication 800-57, Recommendation for Key Management – Part 1: General (Rev. 3) (July 2012), at 66.

47. Beyond the reduction in key size, the mathematic operations used in elliptic-curve cryptography—scalar multiplication on an elliptic curve—is fundamentally simpler than the large-integer exponentiation and modulus operations used in classic Diffie-Hellman. Thus, the computational cost for ECDH is significantly lower, allowing key exchanges to complete quickly, even on low-clock-speed microcontrollers.

48. This makes ECDH particularly well-suited for resource-constrained environments such as SIM cards, and embedded secure elements such as embedded Universal Integrated Circuit Cards (eUICCs)—devices that operate with limited computational resources and memory, but that must still provide robust cryptographic protection. Accordingly, ECDH and other algorithms based on elliptic-curve cryptography have become the preferred choice for secure channel establishment in constrained platforms. For example, as described in further detail below, GlobalPlatform introduced SCP11 as an elliptic-curve-based successor to SCP10.

c. Cryptographic Standards/Requirements

49. Cryptography is only as strong as its weakest component. A

theoretically strong algorithm provides little real security if implemented incorrectly, used with insufficient key lengths, or combined with untested random-number generators, or other components. For this reason, the cryptographic community relies on formal standards to define not only which algorithms are acceptable, but also how they must be implemented, tested, and validated.

(1) NIST

50. In the United States, the National Institute of Standards and Technology (NIST) plays a central role in defining and maintaining cryptographic standards. NIST develops Federal Information Processing Standards (FIPS) and associated Special Publications that govern algorithm selection, key lengths, operational modes, and key-management practices for both federal agencies and private-sector organizations seeking validated security protocols. *See, e.g.*, Ex-1023, NIST SP 800-57, at 15. By 2013, NIST had long been the central authority for cryptographic standards in the United States, and its publications had become the de facto global reference for both government and commercial systems. Indeed, NIST's standards form the basis for compliance regimes in banking, telecommunications, payment systems, and mobile security.

51. By 2013, the following NIST standards were all well-established and publicly available:

- **SP 800-57 – Recommendation for Key Management.** Ex-1023. This publication provides comprehensive guidance on key lifecycles—covering generation, distribution, storage, use, rotation, and destruction—specifying how different classes of keys (e.g., data-encryption keys, key-encrypting keys, authentication keys) should be protected.
- **SP 800-56A – Recommendations for Pair-Wise Key-Establishment Schemes.** Ex-1025. This publication formalizes the various algorithms to be used for key-agreement and key-transport.
- **SP 800-38F – Recommendation for Key-Wrapping Using AES.** Ex-1024. This publication defines how to encrypt or “wrap” one cryptographic key using another key (known as a “Key-Encrypting Key” (KEK) or “Key-Wrapping Key” (KWK)) to ensure its confidentiality during transport or storage. As SP 800-38F explains, “[s]egregating keys from general data” and encrypting such keys with a KEK/KWK “can provide an extra layer of protection” for such sensitive data. Ex-1024, NIST SP 800-38F, at 2.

52. By 2013, these standards collectively defined a coherent framework for secure algorithm selection and key handling. They were widely cited by device manufacturers, telecommunications operators, and standards organizations, such as GlobalPlatform. *See, e.g.,* Ex-1026, GlobalPlatform Card Technology, Secure

Channel Protocol 03, Card Specification v2.2 – Amendment D (“GlobalPlatform SCP03”), at 3.

53. For manufacturers and service providers, adherence to NIST’s framework ensured that cryptographic implementations were both technically sound and regulatorily defensible. Compliance demonstrated that a product’s encryption, key-wrapping, and authentication functions had been designed in accordance with publicly recognized best practices, rather than proprietary or unverified methods.

54. In the context of embedded and mobile security—including SIMs, eUICCs, and other secure elements—these standards provided the blueprint for how cryptographic keys should be generated, exchanged, and protected. As a result, by 2013, a person of ordinary skill in the art would have been fully familiar with NIST’s published key-management and key-protection requirements, and would have understood them as the prevailing and expected baseline for any secure communication or provisioning system.

(2) Key Protection

55. A fundamental principle in NIST’s framework—well established before 2013—is that keys must always be protected at or above the security level of the data they are used to safeguard.

56. Thus, during transport, cryptographic keys must never appear in

plaintext. Instead, they are protected using key-wrapping algorithms (as described in NIST SP 800-38F) or derived securely through key-agreement protocols such as Diffie–Hellman or Elliptic Curve Diffie–Hellman (as described in NIST SP 800-56A). And, during storage, keys must remain encrypted or bound to a specific cryptographic module that enforces secure access controls. *See, e.g.*, Ex-1023, NIST SP 800-57, 73 (“When the symmetric or private key exists external to the cryptographic module, confidentiality protection **shall** be provided either by encryption (e.g., key wrapping) or by controlling access to the key via physical means (e.g. storing the keying material in a safe with limited access.”)

2. Authentication

57. During a secure communication session, a recipient device must be able to confirm that the sending device is genuine and not an impersonator. Authentication ensures that each party can verify the identity of the other and thereby confirm that the messages it is receiving originate from a legitimate source. *See, e.g.*, Ex-1021, Boyd & Mathuria, at 5-8, 15.

58. Authentication may be achieved in multiple ways. In a symmetric authentication method, both parties can prove their authenticity by demonstrating knowledge of the same shared secret. This may be accomplished through the use of a challenge-response protocol: one party sends a challenge; the receiving party

encrypts (or “signs”) the received challenge using its copy of the shared secret key and returns the result; if the challenger can decrypt or verify the response using its own copy of the shared secret key, this proves that the receiving party is a legitimate holder of the secret key.

59. Authentication can also be performed asymmetrically. Such methods typically rely on digital signatures and Public Key Infrastructure (PKI). Each entity possesses a private signing key and a corresponding public verification key. A trusted certificate authority (CA) issues a digital certificate binding the entity’s identity to its public key. *See, e.g.,* Ex-1025, NIST SP 800-56A, at 38-39. During authentication, one party proves its identity by signing a message or challenging with its private key; the recipient verifies that signature using the public key obtained from the certificate. This prevents impersonation and enables scalable trust relationships across large networks.

3. Integrity and Replay Protection

60. Integrity protection ensures that data received has not been altered—either accidentally or maliciously—during transmission. *See, e.g.,* Ex-1021, Boyd & Mathuria, 15. Cryptographic mechanisms such as Message Authentication Codes (MACs) or digital signatures provide this assurance, by allowing the recipient to verify that a message is authentic and unaltered. are used to detect any unauthorized

modifications. A MAC combines a message with a secret key to produce a short authentication tag that accompanies the message; any alteration of the message or tag invalidates verification.

61. A Message Authentication Code (MAC) is a short cryptographic tag derived from both the message and a shared secret key. *See, e.g.*, Ex-1023, NIST SP 800-57, at 35. The sender computes the MAC by applying a cryptographic function to the message together with the secret key. The resulting tag is appended to the message before transmission. *See, e.g.*, Ex-1023, NIST SP 800-57, at 37.

62. When the recipient receives the message and its accompanying MAC, the recipient recomputes the MAC using the same secret key and the received message. If the newly computed tag exactly matches the received tag, the recipient can conclude that (1) the message has not been altered in transit (integrity), and (2) the message was generated by someone possessing the secret key (authenticity).

63. If even a single bit of the message or tag is changed, the recomputed MAC will not match, and the verification fails. This property arises from the cryptographic strength of the underlying function—without the secret key, an adversary cannot feasibly generate a valid MAC for a modified message.

64. Digital signatures serve a similar purpose but use asymmetric cryptography. The sender signs a message with a private key, and the recipient

verifies the signature using the corresponding public key. Digital signatures are typically used when authenticity must be verified without a pre-shared secret (for example, in certificate-based authentication), while MACs are used in symmetric-key systems where both parties already share a common secret.

65. Replay protection complements integrity protection by ensuring that even a valid message—one that was correctly signed or MAC-protected—cannot be maliciously reused or “replayed.” *See, e.g.*, Ex-1021, Boyd & Mathuria, at 8-12. Without replay protection, an adversary could capture a legitimate, authenticated message and retransmit it later to trigger the same action again (for example, re-executing a payment instruction or re-authenticating to a service).

66. Common mechanisms for preventing replay attacks include:

- **Nonces (numbers used once):** A nonce is a unique value (e.g., a random or pseudorandom number) included in each message or session. *See, e.g.*, Ex-1021, Boyd & Mathuria, at 9. Because the nonce is unique, an attacker cannot reuse an old, otherwise valid message—that stale message would contain an outdated nonce and would be rejected. *See, e.g.*, Ex-1021, Boyd & Mathuria, at 22.
- **Timestamps:** A timestamp records when a message was created or sent. The recipient verifies that the timestamp falls within an acceptable time window.

This method is lightweight and easy to implement but relies on both parties maintaining closely synchronized clocks. *See, e.g., See, e.g.,* Ex-1021, Boyd & Mathuria, at 22.

- **Sequence numbers:** Each message is assigned an incrementing counter value, allowing the recipient to detect duplicates or out-of-order messages. This technique provides reliable ordering and replay detection but requires both sides to maintain persistent counters to survive resets or power loss. *See, e.g.,* Ex-1021, Boyd & Mathuria, at 22-23.

67. Together, integrity and reply protection help to ensure that a message is both authentic and fresh, preventing an attacker from modifying or reusing communications to gain unauthorized access or cause unintended actions.

B. GlobalPlatform

68. GlobalPlatform is an international standards organization established in 1999 to define open, interoperable specifications for secure application management on smart cards and embedded secure elements (SEs). *See, e.g.,* <https://en.wikipedia.org/wiki/GlobalPlatform>; *see, also* <https://globalplatform.org/why-globalplatform>. It was founded by a consortium of major payment networks (including Visa and MasterCard), telecommunications operators, and technology companies that recognized the need for a common,

vendor-neutral architecture for secure chips used across industries. *Id.*

1. Secure Elements and Security Domains

69. A smart card or secure element is a tamper-resistant hardware device that securely stores cryptographic keys and executes security-critical operations (e.g., authentication, encryption, or digital signatures).

70. A GlobalPlatform-compliant secure element is organized into one or more Security Domains (SDs)—logical containers that hold cryptographic keys, security policies, and application management privileges. Ex-1006, 20. Each smart card includes an Issuer Security Domain (ISD), which is the “mandatory on-card representative of the Card Administrator.” Ex-1006, 20. It holds the issuer’s master keys, authenticates to remote servers, and establishes secure channels for management operations. Other security domains (Supplemental Security Domains or SSDs) may be created for third-party service providers (e.g., Visa, MasterCard, or a mobile network operator). Ex-1006, 20. Each SSD isolates that provider’s applications and data from others on the same chip, while enforcing access-control policies and cryptographic separation.

71. This multi-domain architecture enables a single card or device to host multiple secure applications—such as a payment app and mobile-network profile—each controlled by its own administrative entity under the same GlobalPlatform

framework.

2. Remote Management and Secure Channels

72. To install, update, or personalize applications, a remote management system must communicate securely with the security domain on the card or device. GlobalPlatform defines Secure Channel Protocols (SCPs) to ensure that these communications are both authenticated and protected from eavesdropping or tampering. Ex-1006, 28.

73. The SCP establishes a mutually authenticated session between the remote server (often called the Card Manager or Off-Card Entity) and the on-card Security Domain. Once a secure channel is established, all commands—such as loading an application, deleting keys, or updating configuration data—are cryptographically protected for confidentiality, integrity, and authenticity. Ex-1006, 28.

74. Over the years, GlobalPlatform has defined several generations of Secure Channel Protocols, each improving upon its predecessors in terms of cryptographic strength and flexibility:

- **SCP01 and SCP02:** In SCP01 and SCP02, both parties (the card and the management server) share static master keys. These protocols authenticate the server to the card and derive session keys for confidentiality (ENC) and

integrity (MAC), from the static keys. In SCP01, the data encryption key (DEK) for decrypting sensitive data sent to the card (e.g., secret or private keys) is used as a static key (Ex-1006, 198), while in SCP02, a session DEK is derived from the static DEK (Ex-1006, 216).

- **SCP03:** SCP03 was based on the existing SCP01 and SCP02 protocols, and introduced to replace 3DES encryption with AES-based cryptography, offering stronger security and improved performance. Ex-1026, GlobalPlatform SCP03, 5.
- **SCP10:** SCP10 marked a shift from purely symmetric cryptography to asymmetric key-agreement mechanisms. Ex-1006, 14.
- **SCP11:** SCP11 introduced Elliptic Curve Cryptography for mutual authentication and key agreement. Ex-1011, 7.

75. Across all SCP versions, GlobalPlatform employs multiple types of session keys, typically including:

- an encryption key (S-ENC) for encrypting data,
- a MAC key (S-MAC) for ensuring integrity and authenticity, and
- a Data Encryption Key (DEK) for protecting sensitive keys and other data transmitted during management operations.

76. The DEK is particularly important when new keys must be injected into

a card—such as during a PUT KEY command. Ex-1006, 264. The DEK ensures that even the key material being provisioned is transmitted in encrypted (wrapped) form. Depending on the SCP version, the DEK may be static (pre-installed on the card and known to the issuer) or session-specific (derived from the secure-channel negotiation).

C. SIMs and eSIMs/eUICCs

77. A Subscriber Identity Module (SIM) is a type of smart card that may be inserted into a mobile device and used to authenticate a subscriber to a mobile network operator (MNO). *See, e.g.*, Ex-1027, 5. A SIM contains a hierarchical file system composed of elementary files (EFs) and dedicated files (DFs), which store network credentials, operator policies, and subscriber information. Ex-1027, 16. Among the key data elements are:

- **International Mobile Subscriber Identity (IMSI):** A unique identifier assigned to each subscriber. *See, e.g.*, Ex-1027, 16.
- **Authentication Key (K):** A secret shared between the SIM and the MNO's authentication center (AuC). Ex-1027, 20. This key is never exposed; it is used within the SIM to compute authentication responses (e.g., RES) to network challenges. Ex-1027, 22.
- **Network Access Applications (NAAs):** Logical applications that execute

authentication algorithms and manage access to the operator's network services. Ex-1027, 55.

- **NAA parameters and network-selection data:** Information defining how the device chooses and prioritizes available networks when roaming.

78. To guide network selection when roaming, the network access parameters stored by the SIM include lists of preferred and forbidden networks. In particular, the SIM stores a Preferred PLMN List, which specifies the networks that the subscriber's home operator prefers the device to use when roaming, and a Forbidden PLMN List, which specifies the networks that the device is forbidden to use when roaming. Ex-1027, 17. Each entry in each of these lists includes a PLMN Identifier, which identifies a specific mobile network operator. *See, e.g.,* https://en.wikipedia.org/wiki/Public_land_mobile_network. The PLMN Identifier includes two numeric components: (1) a Mobile Country Code (MCC) identifying the country in which the network operates, and (2) a Mobile Network Code (MNC) identifying a specific network within that country. *Id.* Together, the MCC + MNC pair forms a globally unique identifier, often written as a six-digit code.

79. Traditional SIMs are operator-locked, meaning the credentials and configuration on the card correspond to a single MNO. To change carriers, the subscriber must physically remove the SIM and insert a new one containing different

keys and network parameters. Ex-1027, 5. This model poses challenges for devices that are embedded, remotely deployed, or difficult to access—such as vehicles, industrial sensors, or IoT modules.

80. To overcome these limitations, the industry introduced the embedded SIM (eSIM) and its underlying hardware platform, the embedded UICC (eUICC). The eUICC is a soldered or embedded secure element that performs the same cryptographic and authentication functions as a traditional SIM but allows the stored operator profile to be remotely provisioned, managed, and replaced without physical card exchange. Ex-1027, 5.

81. The concept originated in machine-to-machine (M2M) deployments, where physical replacement was impractical, and later expanded to consumer devices such as smartphones, tablets, and wearables. The eUICC architecture separates the hardware (the secure element) from the operator profiles it contains. Each profile acts as a digital equivalent of a physical SIM, complete with its own IMSI, authentication key K, PLMN lists, and network-access applications.

82. eSIM operation relies on the secure remote provisioning of these operator profiles to devices. By 2013, the process had begun to establish standards to govern profile provisioning. For instance, by 2013, ETSI had released a Requirements Specification for Smart Cards and Embedded UICCs, which

described the various use cases for smart cards and eUICCs, along with security requirements for remotely provisioning profiles to those devices. *See* Ex-1028, ETSI TS 103 383 v12.1.0, “Smart Cards; Embedded UICC; Requirements Specification (Release 12),” (June 2013) (“TS 103.383”). TS 103.383 describes the use of a “Subscription Manager” to provision profiles to eUICCs. Ex-1028, 6-7. The Subscription Manager consists of two roles: (1) “Subscription Manager – Data Preparation (SM-DP)”; and (2) “Subscription Manager – Secure Routing (SM-SR).” *Id.* The SM-DP “prepares operational and provisioning profiles to be securely provisioned on the eUICC,” and the SM-SR “securely performs functions which directly manage the operational and provisioning profiles on the eUICC,” including delivering the profiles prepared by the SM-DP to the eUICC. *Id.* These entities are illustrated in Figure C.1, which depicts the provisioning environment:

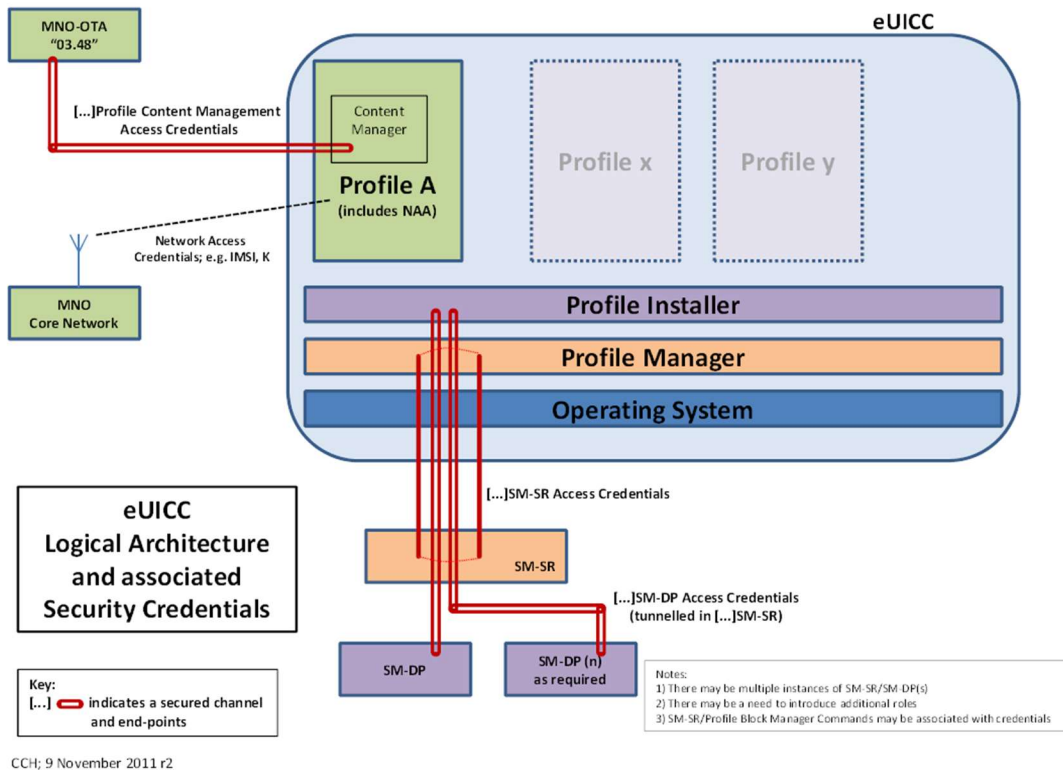
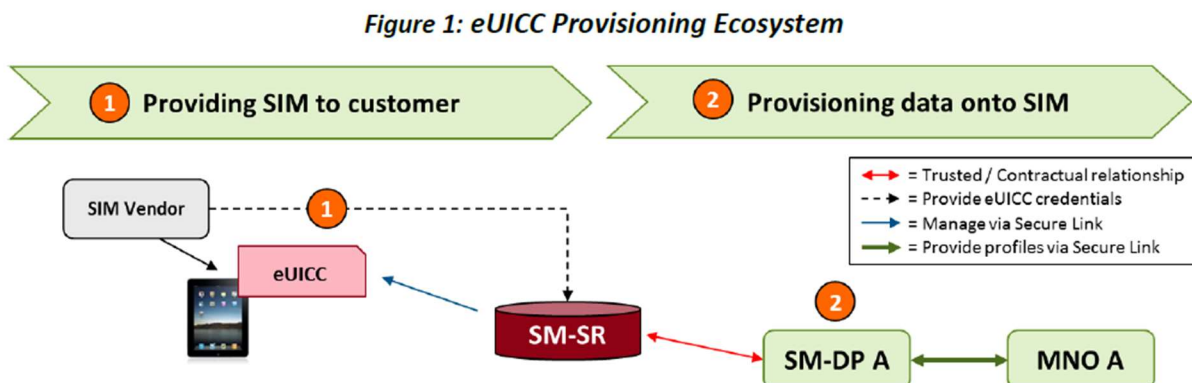


Figure C.1

83. Similarly, in September 2012, Ofcom released a report on “Reprogrammable SIMs,” describing the “eUICC Provisioning Process and Ecosystem.” Ex-1028, 6. Figure 1, reproduced below, illustrates the provisioning environment:



Ex-1027, 7. Like TS 103.383, the Ofcom Report explains that, “[i]n the eUICC ecosystem, a Subscription Manager is required,” with the Subscription Manager role split into the two functions of “Secure Routing (SM-SR) and Data Preparation (SM-DP).” Ex-1027, 6. The Ofcom Report describes the provisioning process as follows:

When an eUICC is manufactured, the eUICC Issuer will load the master keys of the eUICC into the SM-SR database. These keys allow the SM-SR to access the eUICC and download new operator profiles to the eUICC. The SM-SR interconnects with a series of SM-DPs. SMDPs package the MNO SIM credentials (including the IMSI and Ki) for secure download. When the consumer requests an MNO profile, the SM-SR associated with the eUICC requests the profile from the relevant SM-DP and downloads it to the eUICC. Secure communications between the SM-SR and a new eUICC occurs over a “provisioning profile” or “setup bearer”, which is effectively a bootstrap profile that allows the first MNO profile to be provisioned to the eUICC.

Ex-1027, 7.

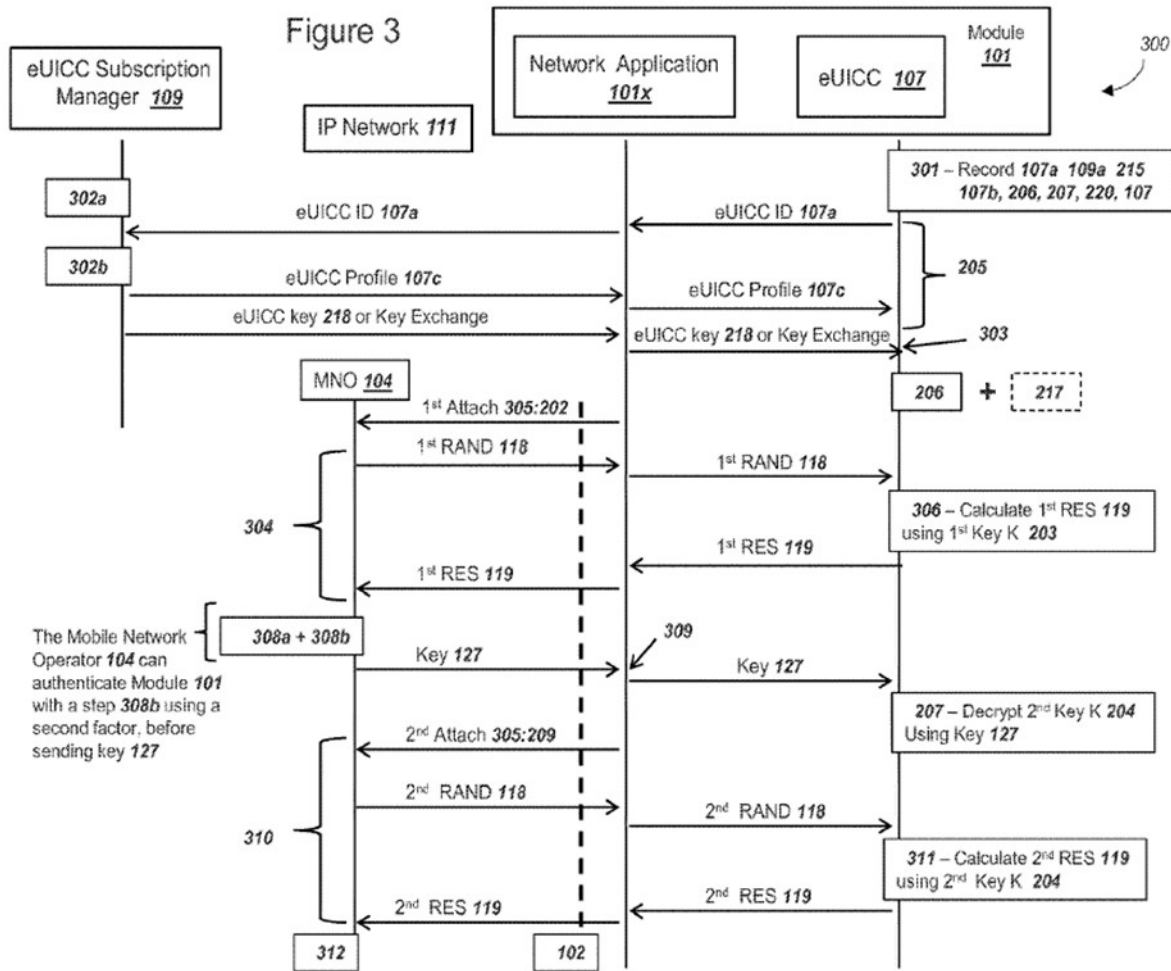
VIII. THE '893 PATENT AND ITS FILE HISTORY

84. The '893 Patent relates to techniques for provisioning an embedded universal integrated circuit card (eUICC) with a profile containing network access credentials. Ex-1001, 3:55-67.

85. As the '893 Patent explains, mobile devices traditionally relied on a “subscriber identity module (SIM) card” or “a related universal integrated circuit card (UICC)” to access a wireless network. Ex-1001, 2:34-45. Each card was supplied by a mobile network operator (MNO) and contained “a pre-shared secret

key K in addition to a set of parameters for a mobile phone or user equipment to connect with the wireless network operated by the MNO.” Ex-1001, 2:45-49. Because each card only allowed access to the network of its associated MNO, changing networks required “swapping the UICC card,” the cost of which “could be prohibitive.” Ex-1001, 3:1-15. This led the industry “to consider an embedded UICC” (eUICC), onto which new MNO network access credentials could be transferred, without physically swapping out the UICC. Ex-1001, 3:16-20.

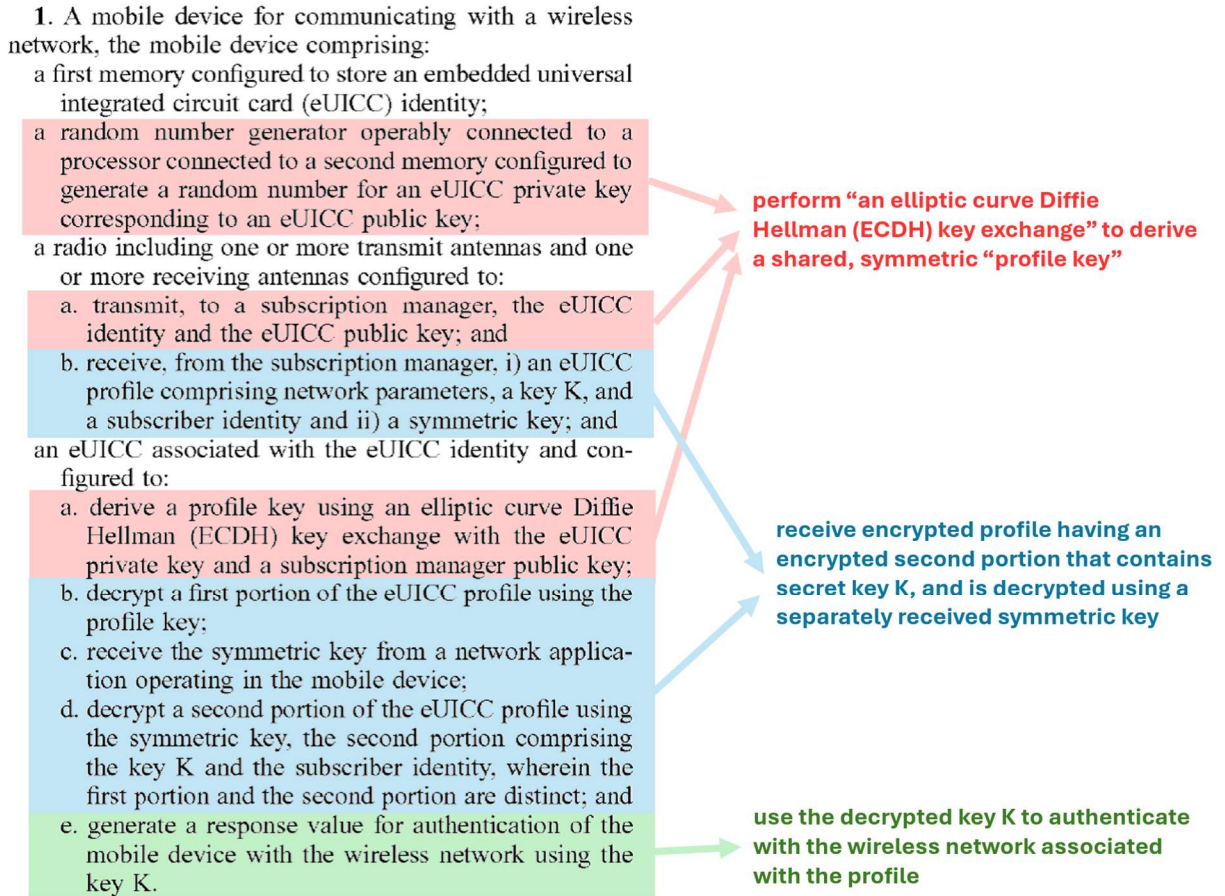
86. The '893 Patent purports to address a security concern with this new provisioning system, where the distribution and release of the profile (including decryption of a secret key K) is under the control of the SM—and not the MNO, which thus loses control over ensuring its secret key K is only provided to authorized devices. Ex-1001, 3:59-64, 35:38-49. To address this issue, the '893 Patent describes a multi-stage provisioning process illustrated in Figure 3, below, splitting control between the SM (for profile delivery) and the MNO (for decryption of secret key K).



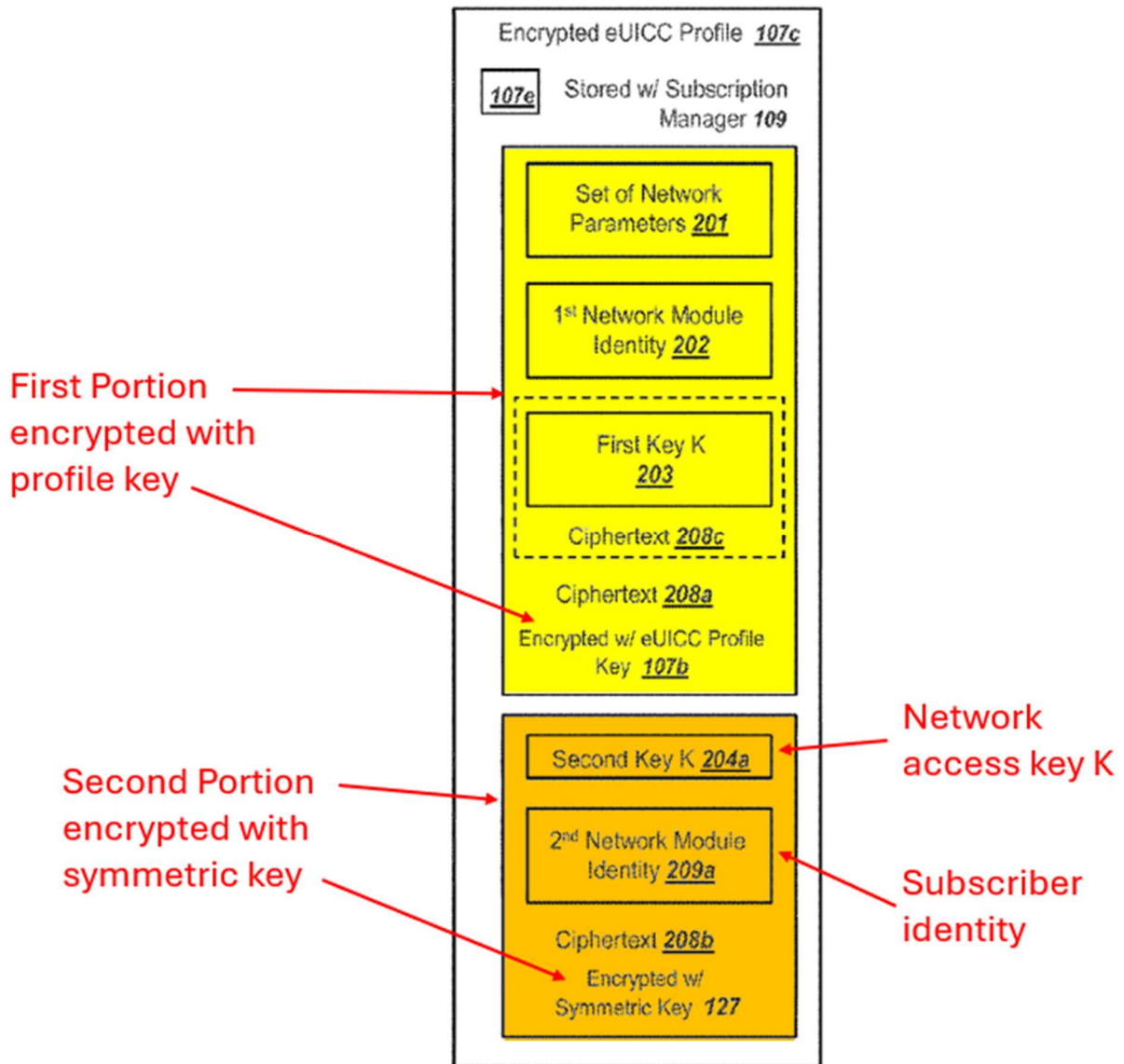
87. The profile 107c delivered to the eUICC by the SM includes two portions—a first portion including a 1st Key K, and a second portion including a 2nd Key K that is encrypted under a symmetric key. *Id.*, 5:35-42, 6:40-42. The eUICC receives this profile from the SM, encrypted under a “profile key” that is known to both the SM and the eUICC. *Id.*, 5:33-40. After decrypting the first portion of the profile using its stored profile key, the eUICC can perform a two-factor authentication with the MNO. *Id.*, 5:45-62; Fig. 3, 308a, 308b. After this second-

factor authentication, the MNO transmits the symmetric key needed to decrypt the second portion of the profile. *Id.*, 6:30-32. Using this symmetric key, the eUICC completes the profile decryption, thereby recovering the second key K, which it can use to gain full access to the network. *Id.*, 6:53-67.

88. Thus, the purpose of the '893 Patent lies in its two-stage eUICC provisioning process, which ensures that the operator—not the subscription manager—is in control of the final release of network credentials. However, Claim 1 (reproduced below) is drafted quite differently, reciting an eUICC performing three conventional processes: (1) participating in an elliptic curve Diffie-Hellman key exchange to generate a shared “profile key” for use in encrypting/decrypting a profile; (2) receiving and decrypting an encrypted profile where the profile is encrypted by two separate keys—the profile key encrypting a first portion of the profile, and a symmetric key encrypting a network access key K and subscriber identity within a second portion of the profile; and (3) using the network access key K to authenticate with the network. Ex-1001, Cl. 1



89. The annotated excerpt of Figure 2a, below, illustrates the two portions of the profile encrypted by two different keys:



90. The '893 Patent was filed December 10, 2021, and claims priority through a chain of 5 continuation applications (now issued patents), the earliest of which was filed December 6, 2013. During prosecution, the Examiner did not issue any substantive prior art rejections, only Section 112 rejections, which the Applicant overcame by amending Claim 1 to recite structure for the claimed “radio” and

“random number generator.” Ex-1004.

IX. PRIORITY DATE

91. I understand that the '893 Patent alleges the earliest priority date of December 6, 2013. However, the claims of the '893 Patent include new matter not disclosed in any of the earlier patents, to which the '893 Patent claims priority.

92. In particular, the common specification makes clear that it is the MNO that sends the symmetric key to the eUICC, to ensure that the MNO maintains control over the release of its network access key K. *See, e.g.*, Ex-1001, 6:30-32, 35:13-38, 35:49-55, 41:48-42:6, 59:37-48, 59:55-65. However, the claims of the '893 Patent recited that the source of the claimed “symmetric key” is the “subscription manager.”

93. I understand from counsel that, when none of the applications in a patent's priority chain provide any written description support for the broad scope of its claims, the patent is entitled to a priority date no earlier than its filing date. Accordingly, the '893 Patent is entitled to a priority date no earlier than December 10, 2021.

94. Ground 6 assumes this later priority date, while the references relied upon or added in Grounds 1-5 are prior art under either priority date.

X. CLAIM CONSTRUCTION

95. For the purposes of this Declaration, I do not believe that any term

requires explicit construction to resolve the issues I present in my Declaration because the prior art discloses or renders obvious the claim limitations under the plain and ordinary meaning of the claim language.

XI. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART

A. Park (Ex-1005)

96. Park was published in the Proceedings of the 2013 International Conference on Availability, Reliability and Security, Regensburg, Germany (pp. 297-303), was cataloged by IEEE, and was publicly available by at least November 7, 2013. Ex-1019. I understand Park qualifies as prior art to the '893 Patent.

97. Park teaches a secure architecture for remotely provisioning MNO profiles onto eUICCs. Ex-1005, 297. Under the eUICC provisioning model newly-introduced by ETSI/GSMA (illustrated in Figure 1, below), the SM was divided into two roles: “SM-DP (Data Preparation)” for generating and encrypting the operator profile, and “SM-SR (Secure Routing)” for transporting and installing the encrypted profile onto the eUICC. *Id.*, 297.

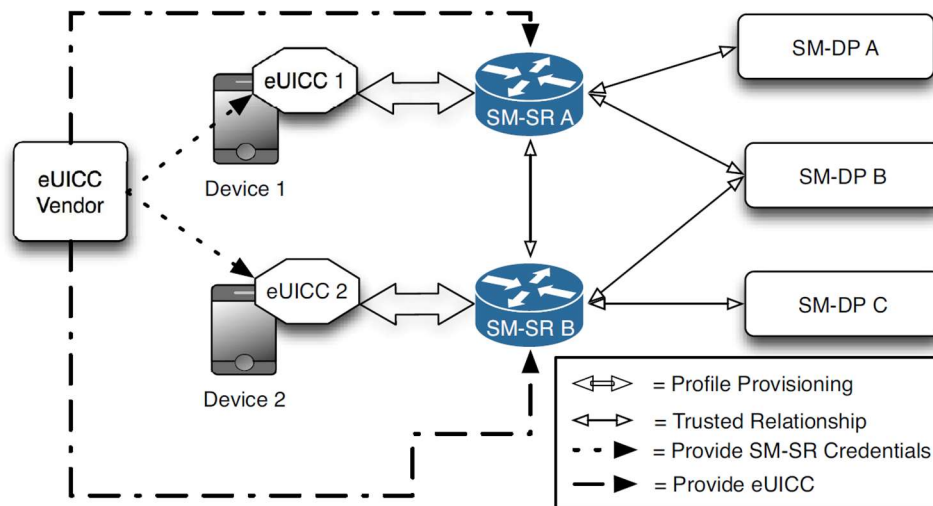


Fig. 1. eUICC Profile Ecosystem

98. This architecture introduced a new trust boundary—because the SM-SR may be operated by an entity distinct from the SM-DP or MNO, it should not be trusted with access to sensitive profile contents. *Id.* To mitigate, Park introduces a Key Agreement Module (KAM), enabling the eUICC and the SM-DP to negotiate mutually compatible profile encryption keys directly, without exposing them to the SM-SR or other intermediaries. *Id.*, 299-301.

B. GlobalPlatform (Ex-1006)

99. GlobalPlatform (GlobalPlatform Card Specification Version 2.2.1) was published in January 2011 and made available that month on the GlobalPlatform Digital Library website. Ex-1018. I understand it qualifies as prior art to the '893 Patent.

100. GlobalPlatform defines a standardized architecture for smart card

security. Ex-1006, 19-20. It defines Secure Channel Protocols (SCPs) to protect communication between the card and an off-card entity, ensuring authentication, integrity, and confidentiality. *Id.*, 28, 112.

101. For the “Secure Channel Protocol ‘10’” (“SCP10”), authentication services are provided “using a Public Key Infrastructure (PKI).” *Id.*, 116. In PKI authentication, each participant has a pair of keys—a private key kept secret and a public key shared with others through a digital certificate issued by a trusted authority. During authentication, each party sends the other a random challenge, which the receiving party signs with its private key and returns to the sender. Ex-1006, 246-47, 266-68, 271. The sender then verifies that signature using the responder’s public key, obtained from its certificate. *Id.* Successful verification proves that the responder is the legitimate holder of the private key and therefore authentic, as shown in Figure F-9, below. *Id.*, 234.

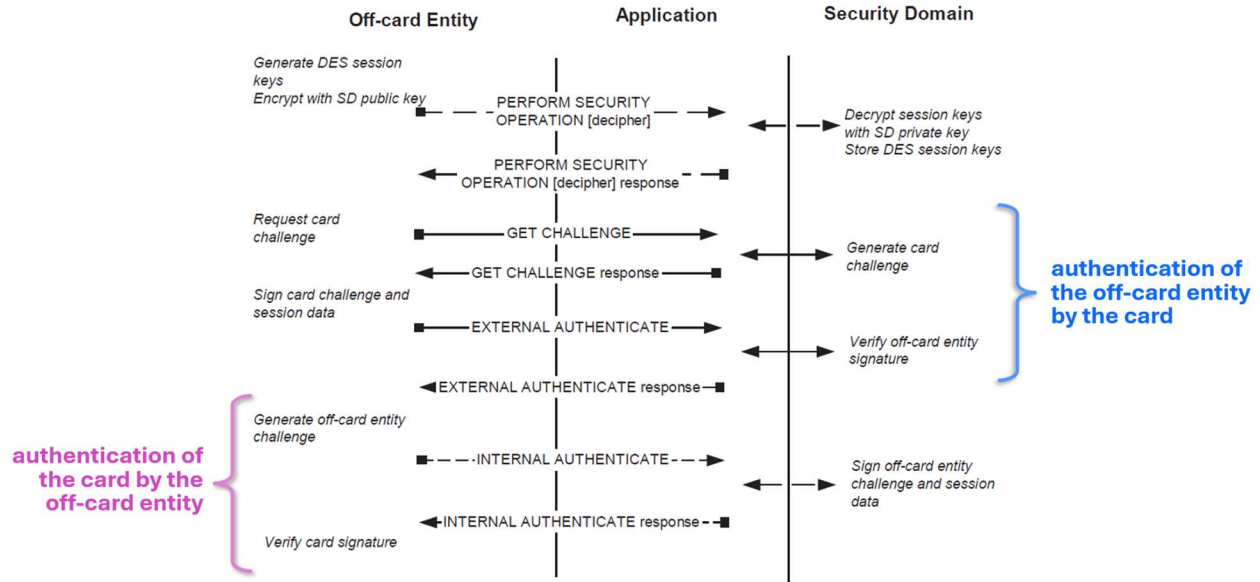


Figure F-9: Entity Authentication Flow

102. For confidentiality, a “Secure Channel session key” encrypts all message data transmitted to and from the card, as illustrated in Figure F-11. *Id.*, 258-59.

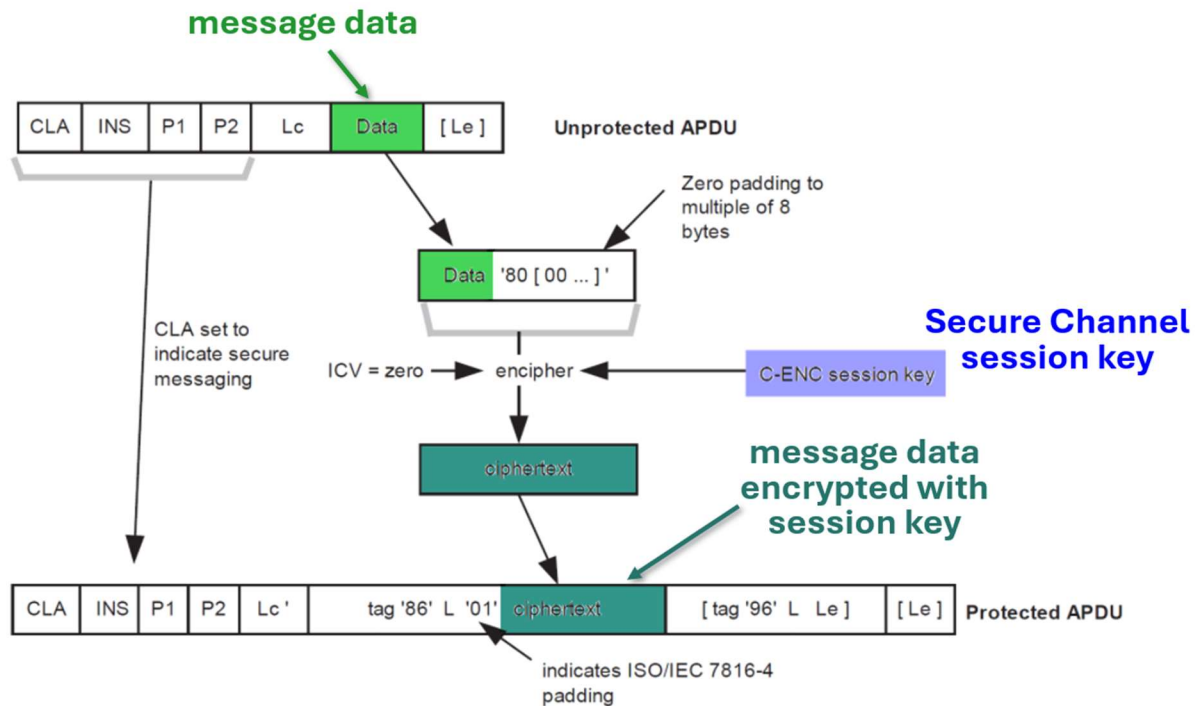


Figure F-11: Secure Messaging: Command message protected for confidentiality

103. GlobalPlatform further imposes an additional level of protection for sensitive data, explaining that “[a] further level of security applies to sensitive data (e.g. secret keys) that shall always be transmitted as confidential data.” *Id.*, 112. This additional encryption “is over and beyond the Current Security Level required for the Secure Channel.” *Id.*, 264. Thus, when “transmitting sensitive data to and from the card”—including “all keys transmitted to a card”—the sensitive data is first encrypted “us[ing] the relevant data encryption session key (DEK)” and then further encrypted with the Secure Channel session key, providing layered protection for these highly sensitive materials. *Id.*, 264.

C. AbiChar (Ex-1007)

104. AbiChar was published in the Proceedings of the 2007 International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST 2007), was cataloged by IEEE, and was made publicly available by at least October 7, 2007. Ex-1018. I understand it qualifies as prior art to the '893 Patent.

105. AbiChar teaches a fast and secure elliptic-curve-based authenticated key agreement protocol (EC-SAKA) for low-power mobile communications. Ex-1007, 1. But as a starting point, AbiChar first considers the standard “Diffie-Hellman key exchange based on EC,” (or ECDH), which AbiChar describes as follows:

Alice selects a random number n_a and computes $Y_a = n_a B$, she 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 $K = n_a Y_b = n_b Y_a = n_a n_b B$.

Ex-1007, 3. The derived key K in AbiChar corresponds to the profile key derived in Claim 1 of the '893 Patent.

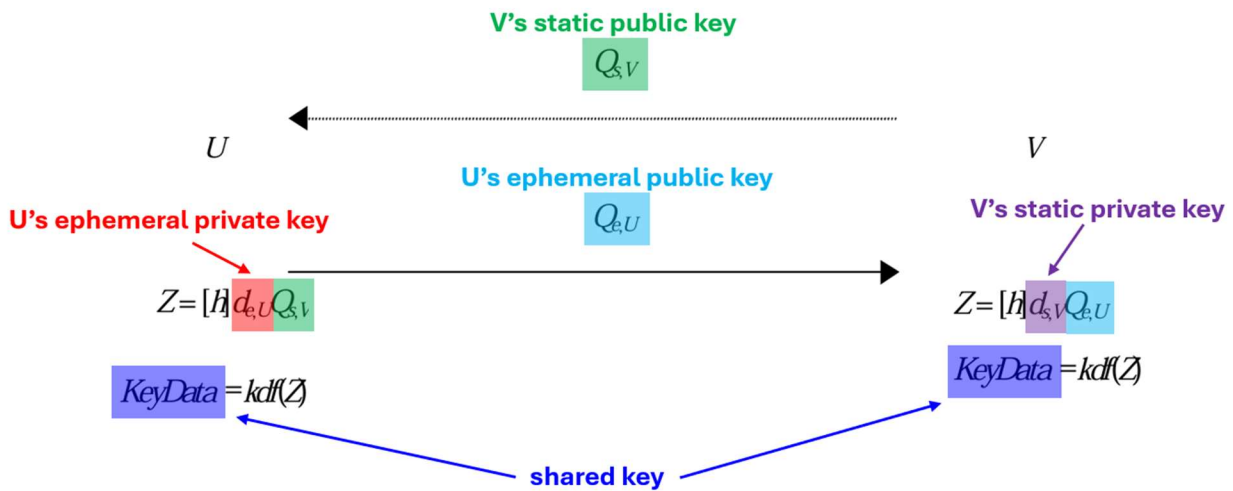
D. X9.63-Overview (Ex-1008)

106. X9.63-Overview (ANSI X9.63 Overview, Key Agreement and Key Transport Using Elliptic Curve Cryptography) was authored by Simon Blake-Wilson in 1999 and was publicly available in 2000. Ex-1015. I understand it

qualifies as prior art to the '893 Patent.

107. X9.63-Overview provides an overview of the ANSI X9.63 standard, which “[s]pecifies key agreement and key transport schemes using elliptic curve cryptography.” Ex-1008, 3. Among the schemes described in X9.63-Overview is the one-pass Diffie-Hellman key-agreement protocol, illustrated below:

ANSI X9.63 - 1-Pass DH



Id., 12. In this protocol, an entity (U) generates an ephemeral public/private key pair ($Q_{e,U}$, $d_{e,U}$) (*id.*), by randomly selecting an ephemeral private key ($d_{e,U}$) (*id.*, 7). Entity U then sends its ephemeral public key ($Q_{e,U}$) to the other entity (V). *Id.*, 12. Each entity then computes a shared secret Z using its own private key ($d_{e,U}$ for U, and $d_{s,V}$ for V) and the public key received from the other entity ($Q_{s,V}$ for U, and $Q_{e,U}$ for V). *Id.* That shared secret is then input into a key-derivation function (KDF) to

produce a **shared key (KeyData)**. *Id.* The scheme is termed “one-pass” because it requires only one transmission of a public key: U sends its ephemeral public key to V, while the V’s static public key is pre-established and known to U, eliminating the need for a reciprocal exchange. The resulting KeyData is substantially similar to AbiChar’s K and the profile key derived in Claim 1.

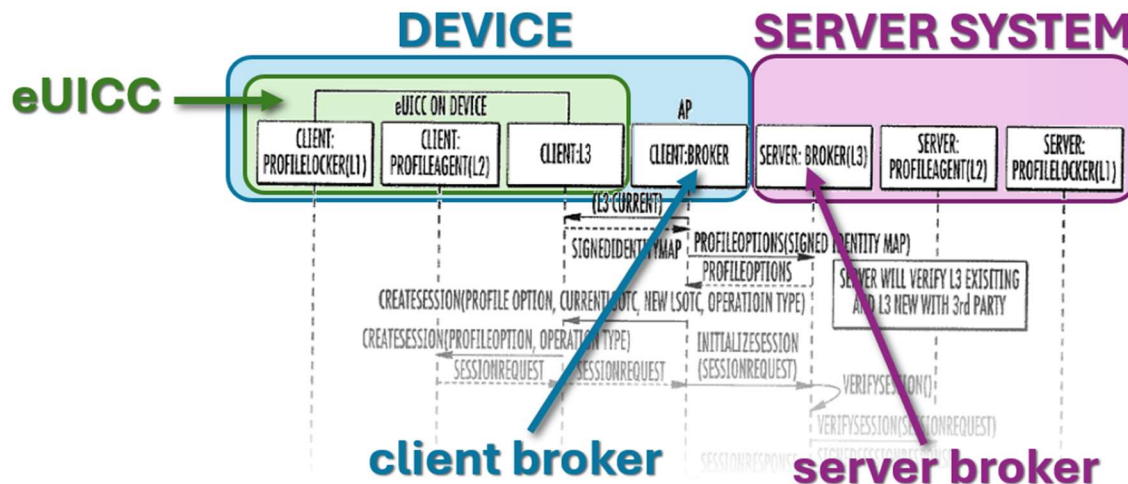
E. Haggerty (Ex-1012)

108. Haggerty (U.S. Patent Pub. No. 2013/0227646) filed February 14, 2013, was published August 29, 2013. I understand it is prior art to the ’893 Patent.

109. Haggerty teaches methods for secure large-scale distribution of electronic access clients, such as eSIMs that are to be stored on eUICCs. Haggerty explains that, although eSIM data are encrypted, certain metadata fields remain in plaintext for operational use, such as the Integrated Circuit Card Identifier (“ICCID”). Ex-1012, ¶84. These plaintext identifiers allow devices and servers to identify and manage profiles without decrypting the full payload. Ex-1012, ¶¶54, 105;

110. In Haggerty’s system, the device on which the eUICC is installed includes a “device broker” for passing communications between the eUICC and the remote provisioning infrastructure, such as session requests and profile download messages. Ex-1012, ¶¶97, 104. The provisioning server system includes a “server

broker” for receiving those requests and relaying them to backend components (e.g., the profile agent and profile locker) that handle cryptographic verification and personalization of the eSIM, as illustrated in Figure 9, excerpted below. Ex-1012, ¶¶105, 108.



111. Haggerty teaches that “the communication between the device broker and the server broker” may use a “a security protocol (e.g., transport layer security (TLS))” to “prevent capture and replay attacks.” Ex-1012, ¶100.

F. Pierce (Ex-1013)

112. Pierce (U.S. Patent Pub. No. 2009/0323967) filed June 30, 2008 and published December 31, 2009. I understand it is prior art to the '893 Patent.

113. Pierce teaches a method for “producing cryptographic keys for use by an embedded processing device within a manufactured product,” using a “pseudo

random number generator” (PRNG). Ex-1013, Abstract. As Pierce explains, generating “strong cryptographic keys typically involves the use of random data that cannot be later discerned or uncovered.” Ex-1013, ¶3. For this purpose, “general purpose computers such as desktop, laptop, and notebook personal computers” typically include hardware/software for generating “entropy data” for use in randomly seeding PRNGs.² *Id.* Embedded processing devices, however, “typically do not have [such] entropy hardware or software engines.” Ex-1013, ¶3. Accordingly, Pierce teaches collecting entropy data from sources that are naturally available to such embedded devices, such as sensor outputs, timing data (e.g., GPS time), unique identifiers, and/or transient events on the communication bus, to generate seed values for the PRNG. Ex-1013 ¶¶35–38, Fig. 3.

G. Konstantinou (Ex-1014)

114. Konstantinou (U.S. Patent Pub. No. 2010/0267383) published on October 21, 2010. I understand Konstantinou is prior art to the ’893 Patent.

115. Konstantinou teaches that UICC profile network parameters include mobile country code (MCC) and mobile network code (MNC) values. Ex-1014 ¶¶31,

² A seed is the initial input that determines the sequence of numbers a pseudo-random number generator will produce; without a sufficiently random seed, the resulting “random” numbers—and thus the cryptographic keys derived from them—could be predicted or reproduced.

158, 162.

H. Nix175 (Ex-1016)

116. Nix175 (U.S. Patent No. 9,100,175) was filed on December 6, 2013 and published on August 4, 2015. If the challenged patent is not entitled to its priority claim as discussed in Section **Error! Reference source not found.**, Nix175 is prior art to the '780 Patent under 35 U.S.C. §102(a)(1). Ex-1016, cover. Nix175 shares a specification with the challenged patent.

XII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS

A. Ground 1: Claims 1, 3, 6-11, 13-17 are obvious over Park in view of GlobalPlatform and AbiChar

1. A POSITA would have been motivated to combine Park's teachings with GlobalPlatform's and AbiChar's teachings and would have had a reasonable expectation of success.

117. A POSITA would have been motivated to combine Park's secure profile provisioning architecture (SPA) with GlobalPlatform's procedures for sensitive data encryption, and AbiChar's elliptic curve Diffie-Hellman (ECDH) key agreement protocol, to yield a complete, secure, and efficient architecture for eUICC profile provisioning, as detailed below.

a. A POSITA would have turned to GlobalPlatform for implementation-level details of Park's security requirements.

118. Park's Secure Profile Provisioning Architecture (SPA) defines an

architectural framework for secure eUICC profile provisioning but intentionally leaves implementation-level details to the practitioner. Ex-1005, 299-303. Park describes the high-level processes by which profiles are generated and securely delivered from the SM-DP to the eUICC and recognizes that “diverse key agreement protocols, secure communication protocols and cryptographic algorithms can be applied depending on the circumstances around the eUICC.” *Id.*, 303. Moreover, Park expressly recognizes the GlobalPlatform Card Specification (GP) and its Secure Channel Protocol (SCP) as the “de facto standard for profile provisioning,” and presents the disclosed SPA as building on and extending GlobalPlatform (which Park refers to as “GP SCP”). Ex-1005, 299. Park makes clear that the fundamental security requirements of its SPA—such as mutual authentication between the SM-DP and the eUICC, and confidentiality of the transmitted profile—are not newly defined, but instead derived directly from GlobalPlatform. In particular, Park expressly builds upon Global Platform, explaining that SPA’s security requirements “should be guaranteed in accordance with the requirements in [5],” where reference [5] corresponds to GlobalPlatform. *Id.*, pp.299, 303.

119. Thus, a POSITA would have looked to GlobalPlatform to understand how to implement Park’s security requirements. GlobalPlatform’s Secure Channel Protocols—including SCP10—were well-known, widely deployed, and thoroughly

vetted mechanisms for providing mutual authentication and confidentiality in smart-card provisioning. Accordingly, incorporating GlobalPlatform's encryption procedures into Park's architecture, would (1) directly satisfy Park's stated requirements, (2) align with the standard that Park's SPA builds upon, and (3) leverage mature, well-tested, and industry-accepted methods rather than devising new ones.

120. In Park's SPA, the eUICC's Key Agreement Module (KAM) is used to generate a shared "SM-DP credential (key_{SM-DP})" for encrypting and decrypting the profile transmitted from the SM-DP to the eUICC. Ex-1005, 300, 302. In the combined Park-GlobalPlatform system, this SM-DP credential (key_{SM-DP}) would serve the same function as GlobalPlatform's Secure Channel session key—providing confidentiality for data exchanged between an off-card entity and the card. Ex-1005, 302; Ex-1006, 258-59. As Park explains, the SM-DP uses the SM-DP credential (key_{SM-DP}), together with "selected cryptographic algorithms" agreed upon by the parties during a capability negotiation process, to encrypt the profile that it sends to the eUICC. Ex-1005, 300. GlobalPlatform provides those established encryption procedures.

121. As explained in Section XI.B, when transmitting message data that includes sensitive data, such as secret/private keys, SCP10 first encrypts the

sensitive data “us[ing] the relevant data encryption session key (DEK)” and then further encrypts the message data with the secure channel session key, providing layered protection for these highly sensitive materials. Ex-1006, 264, 258 (explaining that the session key, C-ENC, is used to encrypt “the command data field of the command message being transmitted to the card,” which “includes any data within the data field that has already been protected for another purpose *e.g. secret or private keys encrypted with the data encryption session key*”); This configuration is illustrated below in annotated Figure F-11.

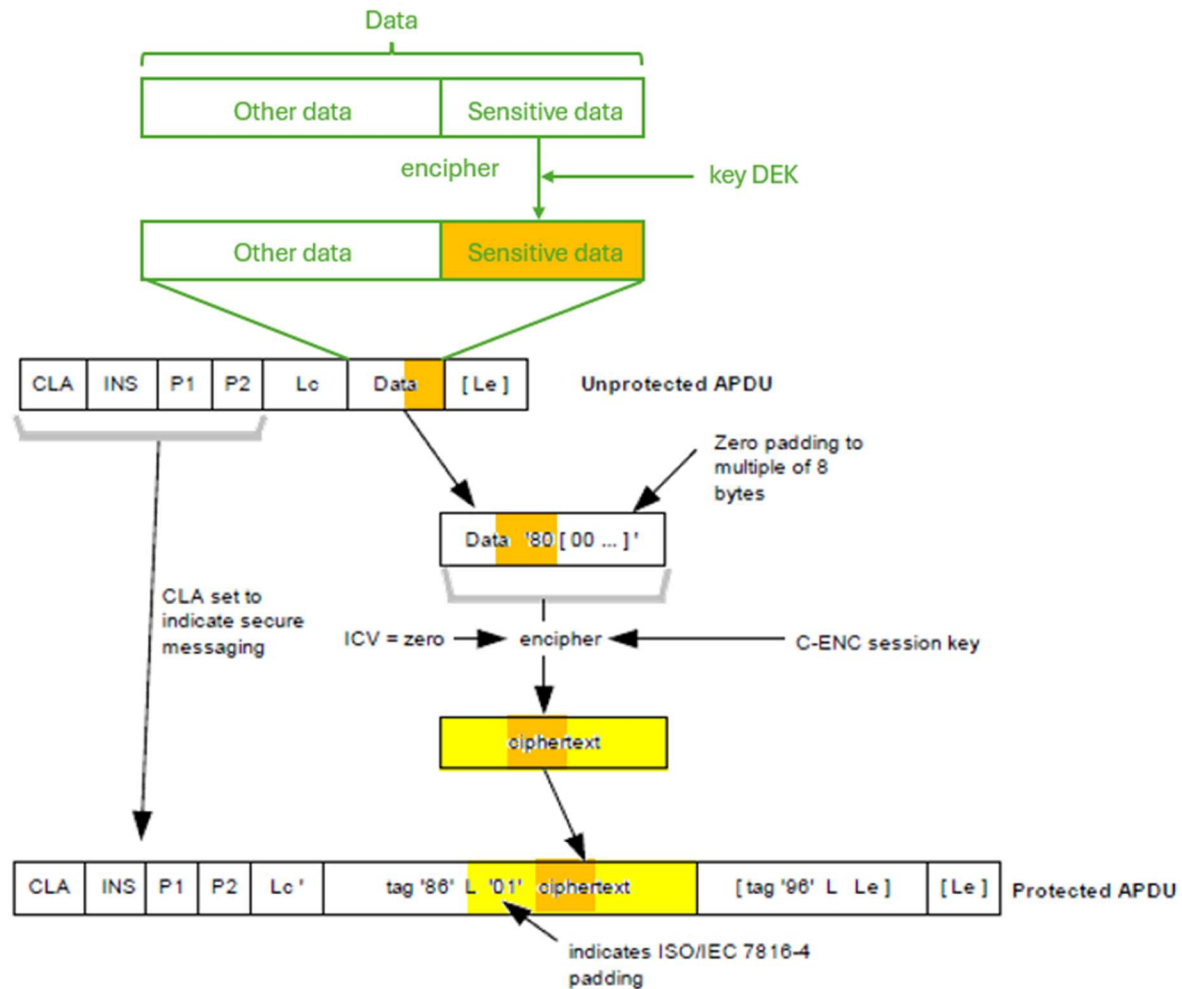


Figure F-11: Secure Messaging: Command message protected for confidentiality

122. Park similarly explains that eUICC profiles contain “security-sensitive information,” including “keys” used for accessing a mobile network. Ex-1005, 297. A POSITA would have recognized that applying GlobalPlatform’s DEK-based inner encryption to these elements of Park’s profile would provide an additional layer of protection for the most sensitive data within the profile, ensuring that even if the outer session-key encryption were compromised, the critical key materials could

remain protected. Thus, in the combined Park-GlobalPlatform system, profiles sent from the SM-DP to the eUICC would include an inner layer of encryption for sensitive materials like keys (using GlobalPlatform's DEK key), as well as an outer layer of encryption using Park's SM-DP Credential (key_{SM-DP})/GlobalPlatform's session key.

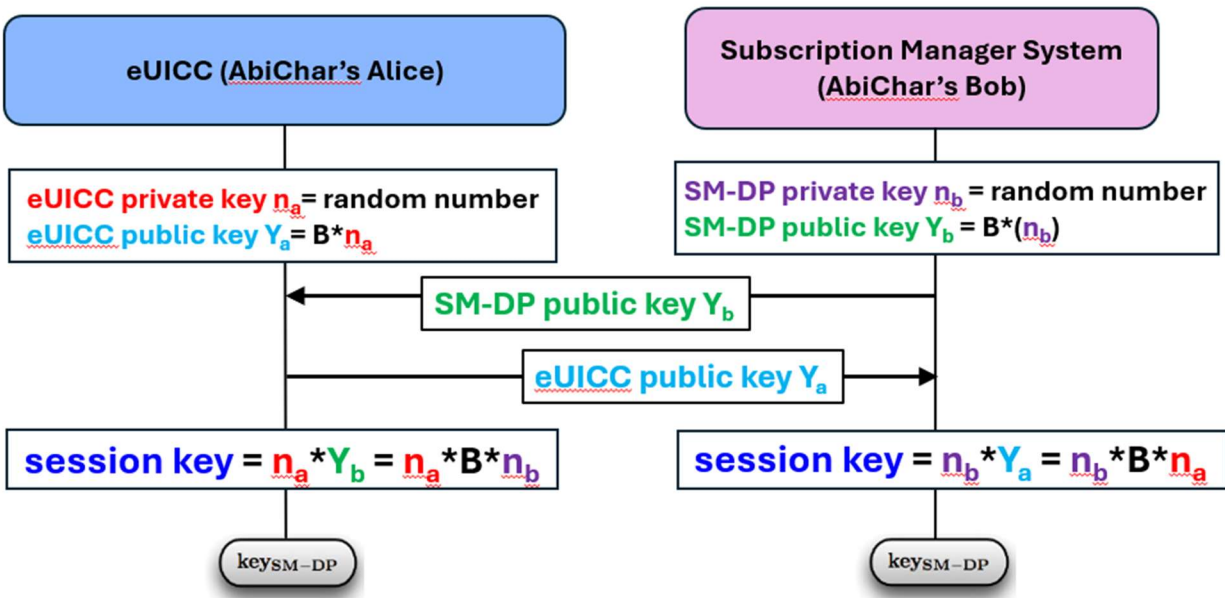
b. A POSITA would have understood that Abi-Char's elliptic curve Diffie-Hellman key agreement could be implemented within Park's SPA to provide an efficient asymmetric key-exchange mechanism

123. Park expressly defines its architecture in terms of asymmetric key exchange, describing a Key Agreement Module (KAM) within the eUICC that "provides the asymmetric algorithm-based key agreement protocol" used to derive a shared SM-DP credential for encrypting the eUICC profile. Ex-1005, 301. However, Park does not specify the particular algorithm to be used, instead explaining that "diverse key agreement protocols ... can be applied depending on the circumstances around the eUICC." Ex-1005, 303.

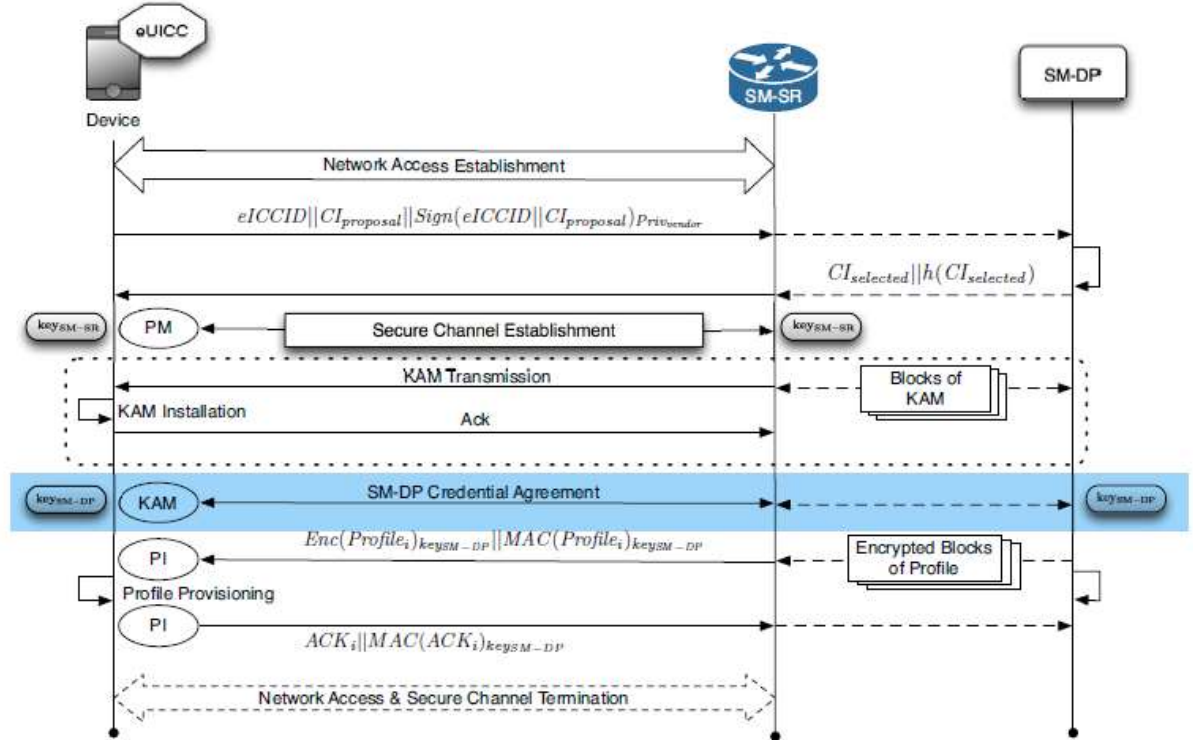
124. A POSITA would have recognized that the elliptic-curve-based Diffie-Hellman key agreement protocol described in AbiChar provides an efficient asymmetric key exchange technique suitable for Park's architecture. Abi-Char explains that elliptic-curve cryptography "can significantly reduce the computation and storage overhead" of public-key operations, making it particularly well-suited

for mobile and smart-card devices with limited processing resources. Ex-1007, 1. AbiChar's EC-SAKA protocol is built upon a standard elliptic-curve Diffie-Hellman (ECDH) key-agreement exchange, during which each party generates random private keys (n_a and n_b respectively), generates public keys from their private keys ($Y_a = n_a B$ and $Y_b = n_b B$, respectively), and then exchanges public keys. *Id.*, 3. Each party then calculates a shared key using its private key and the other party's public key ($K = n_a Y_b = n_b Y_a = n_a n_b B$). *Id.*

125. A POSITA would thus have recognized that incorporating AbiChar's standard Diffie-Hellman scheme within Park's flexible, asymmetric framework, would provide a secure, lightweight, and computationally efficient mechanism for establishing the SM-DP credential (key_{SM-DP}) between the eUICC and the SM-DP. By mapping AbiChar's "Alice" to the eUICC and "Bob" to the SM-DP, the ECDH key exchange between the eUICC and SM-DP can be illustrated as follows:



126. The ECDH exchange occurs in the Park-GlobalPlatform combination after the KAM transmission and installation (which would include ECDH parameters, such as the elliptic curve) and result in the establishment of $\text{key}_{\text{SM-DP}}$ in the “SM-DP Credential Agreement” step:



2. Independent Claim 1

- a. **Element 1[pre]: A mobile device for communicating with a wireless network, the mobile device comprising:**

127. Park in view of GlobalPlatform and AbiChar (“Park-GlobalPlatform-AbiChar”) teaches the preamble, to the extent it is limiting.

128. Park teaches a “secure profile provisioning architecture” for “[e]mbedded UICC (hereafter eUICC),” which Park explains is “a new form of UICC (Universal Integrated Circuit Card)” that is “soldered into a device during manufacturing.” Ex-1002, 297. As illustrated in Figure 5, below, Park teaches that the device housing the eUICC may be a mobile device, such as a smartphone:

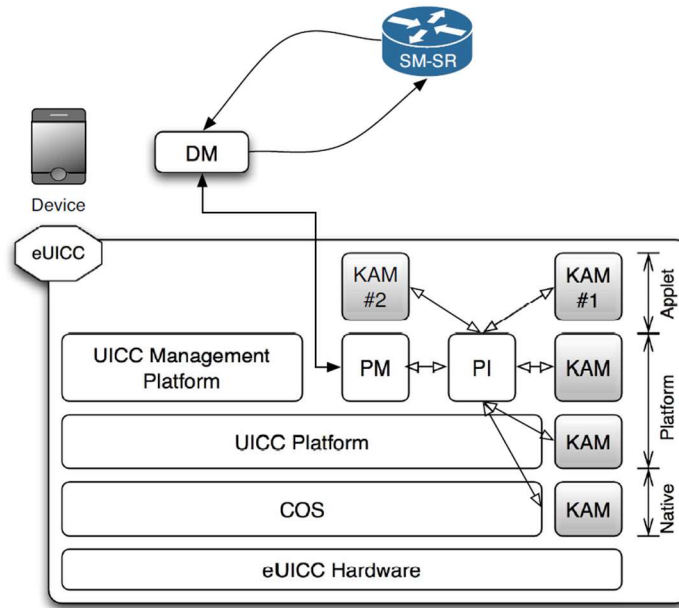


Fig. 5. Internal Architecture of eUICC

129. As Park explains, unlike a traditional UICC that is pre-provisioned by a single MNO, an eUICC must download a profile from a Subscription Manager in order to access an MNO's cellular network. Ex-1005, 297-98. As illustrated in Figure 4, below, downloading a new profile is performed via wireless communications between Park's eUICC-installed device and the subscription manager:

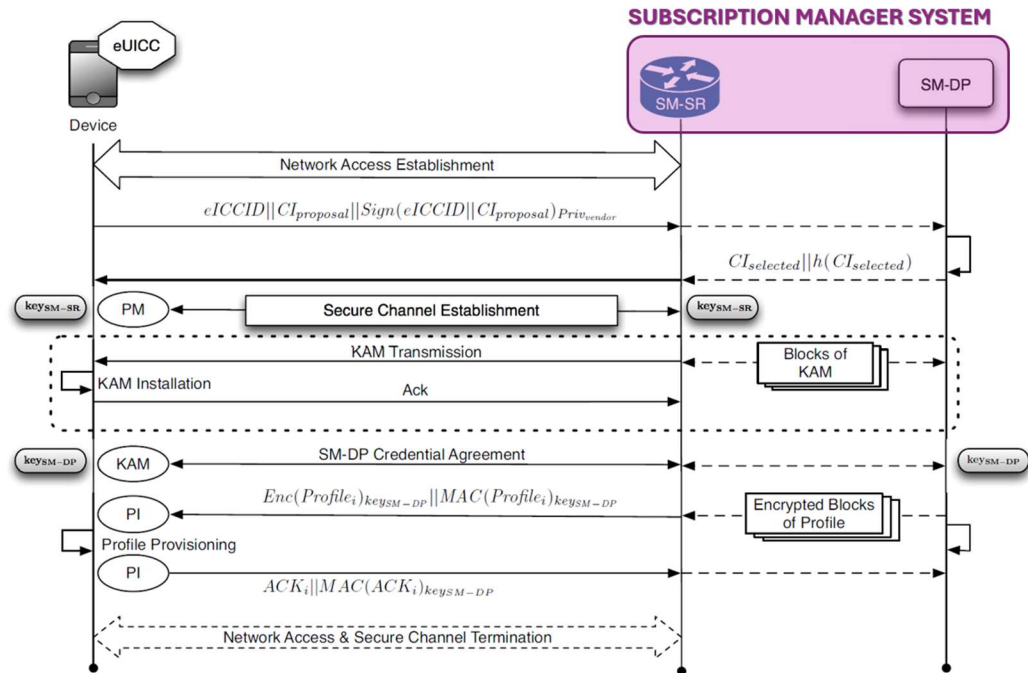


Fig. 4. Secure Profile Provisioning Procedure

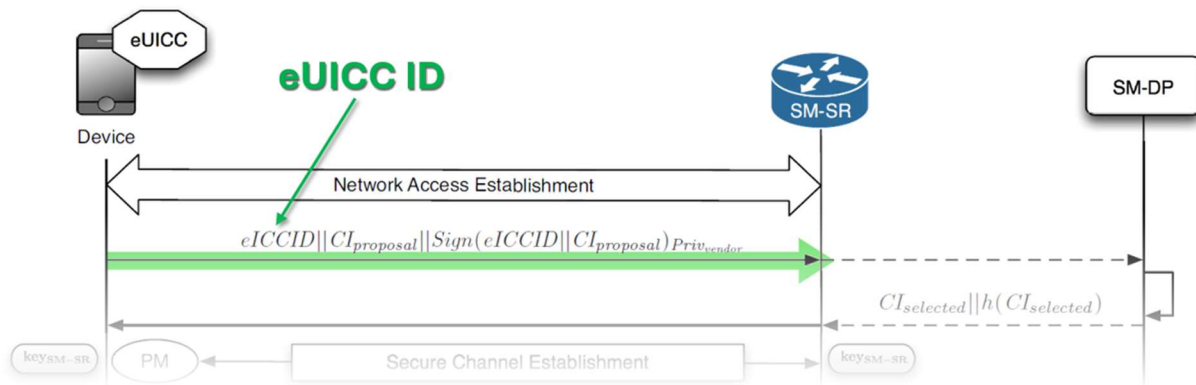
130. Specifically, Park teaches that the eUICC wirelessly communicates with the SM-SR either by: (1) accessing a mobile network “using [a] Provisioning Profile” that was pre-provisioned on the eUICC; (2) “access[ing] SM-SR via WiFi”; or (3) using an already “provisioned profile for accessing [a] mobile network.” Ex-1005, 300. Thus, Park teaches “[a] mobile device for communicating with a wireless network,” as claimed.

- b. Element 1[a]: a first memory configured to store an embedded universal integrated circuit card (eUICC) identity;**

131. Park-GlobalPlatform-AbiChar teaches Element 1[a].

(1) eUICC “identity”

132. Park teaches that, after establishing a communication channel with the SM-SR, the eUICC and SM-DP engage in a capability negotiation, during which they establish the key exchange protocol to be used by the parties. Ex-1005, 301. During this negotiation, the eUICC sends a message to the SM-SR that includes “the eUICC ID (*eICCID*),” as illustrated in Figure 4, excerpted below. Ex-1005, 300.



(2) “a first memory” storing the eUICC ID

133. Park teaches that the eUICC includes “eUICC Hardware,” together with “software running in the eUICC,” as illustrated in Figure 5, below. Ex-1005, 299.

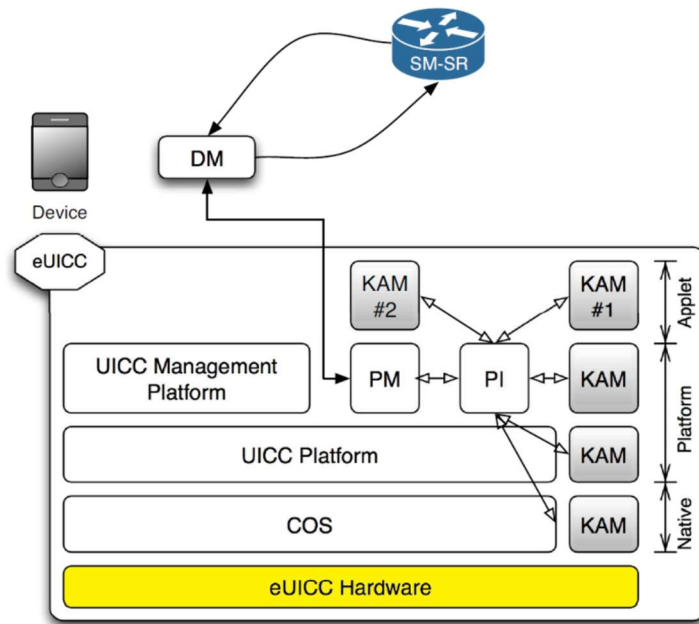


Fig. 5. Internal Architecture of eUICC

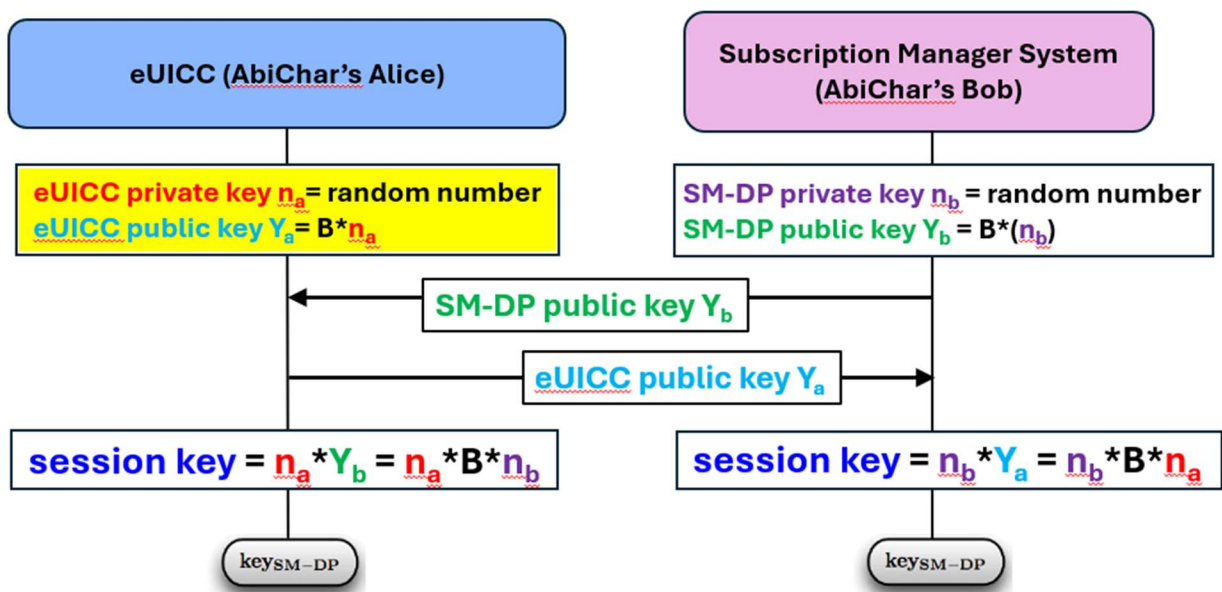
134. A POSITA would have understood that this “eUICC Hardware” includes memory. Indeed, Park explicitly references the memory available to the eUICC, explaining that different eUICC “can be different from each other with respect to ... available memory capacities.” *Id.*, 301. A POSITA would have recognized that the eUICC ID would be stored in this eUICC memory (claimed “first memory”), so that it can be retrieved by the eUICC and used during the provisioning process, as illustrated in Figure 4, above. Ex-1005, 300. Moreover, a POSITA would have preferred that this first memory be non-volatile so the eUICC ID is retained through power cycles.

- c. Element 1[b]: a random number generator operably connected to a processor connected to a second memory configured to generate a random number for an **eUICC private key** corresponding to an **eUICC public key**;

135. Park-GlobalPlatform-AbiChar teaches Element 1[b].

- (1) generating “a random number for an **eUICC private key** corresponding to an **eUICC public key**” using a “random number generator”

136. During Park’s profile-provisioning procedure, the eUICC and SM-DP engage in an “asymmetric algorithm-based key agreement protocol” to each “dynamically [] generate the SM-DP Credentials (key_{SM-DP}).” Ex-1005, 300-301. As explained in Section YY, a POSITA would have been motivated to use AbiChar’s ECDH key agreement for this protocol, resulting in the key exchange illustrated below:



137. As taught by AbiChar, in the combined Park-GlobalPlatform-AbiChar system, the eUICC generates **private key n_a (claimed “eUICC private key”)** as a random number. Ex-1007, 3. The eUICC then calculates a **corresponding public key $Y_a = B * n_a$ (claimed “second eUICC public key”)** from **private key n_a** , using elliptic-curve scalar multiplication, with elliptic curve point B . Ex-1007, 3; A POSITA would have understood that the eUICC would include a random number generator—e.g., within, or coupled to, Park’s KAM software—for use in generating the random number for **private key n_a** .

(2) the random number generator “operably connected to a processor connected to a second memory”

138. Park teaches that the eUICC includes “eUICC Hardware,” together with “software running in the eUICC,” which includes the “Key Agreement Module (KAM),” as illustrated in Figure 5, below. Ex-1005, 299.

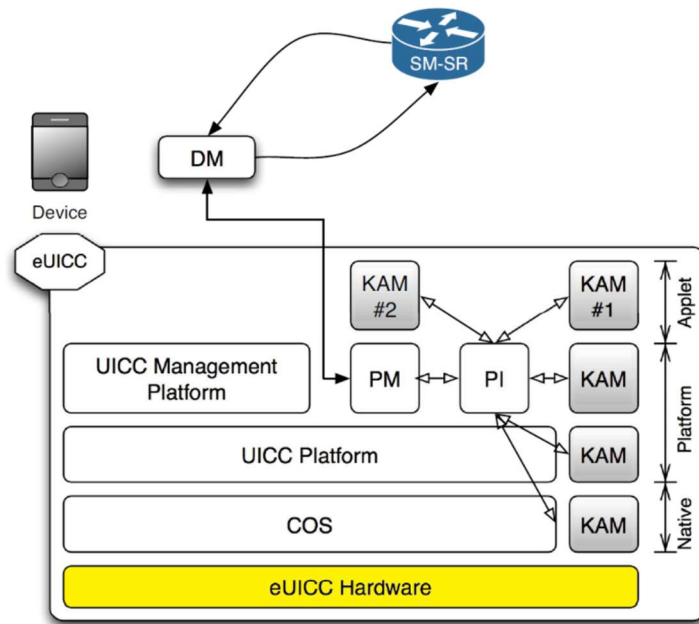


Fig. 5. Internal Architecture of eUICC

139. A POSITA would have understood that the KAM software, together with the random number generator it employs, would execute within the secure environment provided by the eUICC, on one or more processors included within the eUICC hardware—i.e., the random number generator would be “operably connected” to these processor(s), as claimed. Indeed, Park specifically identifies an “(e)UICC” offered by STMicroelectronics that includes a “Crypto Co-Processor,” which a POSITA would have recognized as a well-known component used to execute cryptographic functions, including random number generation, within secure elements such as an eUICC. Ex-1005, 302-03;

140. As explained for Element 1[a], Park also teaches that the eUICC

hardware includes memory. While the first memory for storing the eUICC ID (as discussed for 1[a], above, is likely non-volatile memory, it may be preferable that the random number used to generate the private key n_a not be retained through power cycles to reduce the risk of compromise, in which case, the memory would be a second (volatile) memory, separate from that used to store the eUICC ID. A POSITA would have understood that the random number generator would be connected to that second memory because the random numbers it produces—including random **private key n_a** , generated during the key agreement process—must be stored, so that they can subsequently be accessed and used by the KAM.

d. Element 1[c]: a radio including one or more transmit antennas and one or more receiving antennas configured to:

141. Park-GlobalPlatform-AbiChar teaches Element 1[c].

142. As explained for Element 1[pre], Park teaches that the eUICC wirelessly communicates with the SM-SR during the profile provisioning process, using a “mobile” or “WiFi” network and describes “profiles being transmitted to eUICC.” Ex-1005, 299-300. For this purpose, Park teaches that the eUICC includes a “Device Module (DM)” — “software running in the Device to perform the roles of the intermediary between eUICC and SM-SR,” as illustrated in Figure 5, below. Ex-1005, 299.

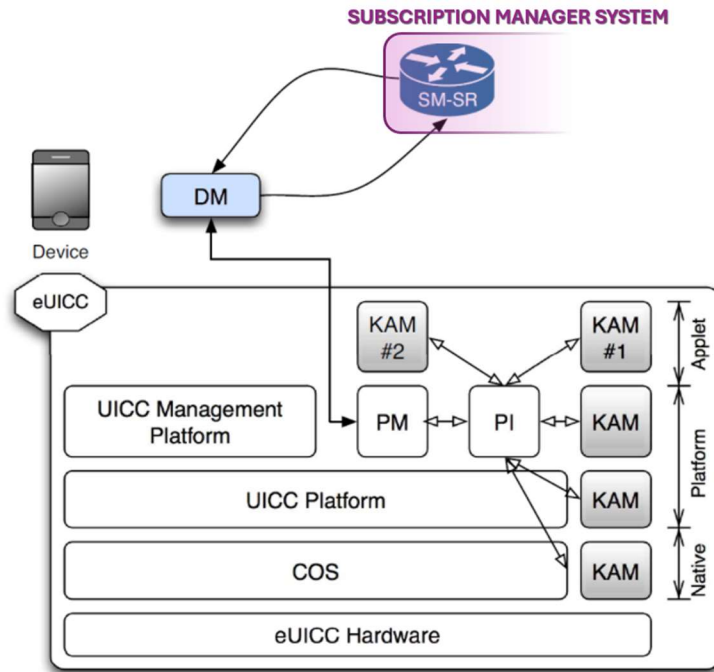


Fig. 5. Internal Architecture of eUICC

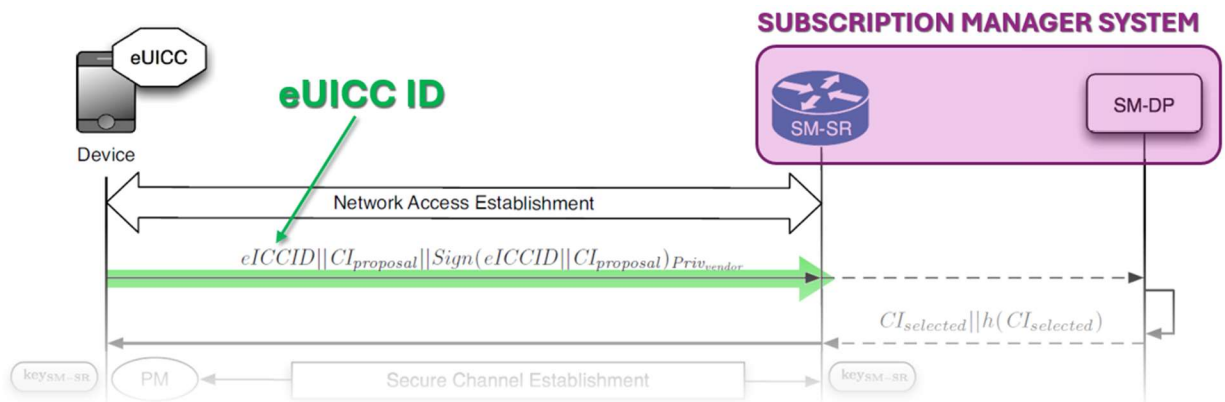
143. A POSITA would have understood that Park's DM would necessarily interface with transmit and receive antenna(s) to enable those wireless communications with the SM-SR. Indeed, it would have been well-known to a POSITA that conventional smartphones—such as the one illustrated in Park's Figure 5, above—include integrated transmit/receive antenna(s) for wireless communications over cellular and WiFi networks.

- e. **Element 1[c][1]: a. transmit, to a subscription manager, the eUICC identity and the eUICC public key; and**

144. Park-GlobalPlatform-AbiChar teaches Element 1[c][1].

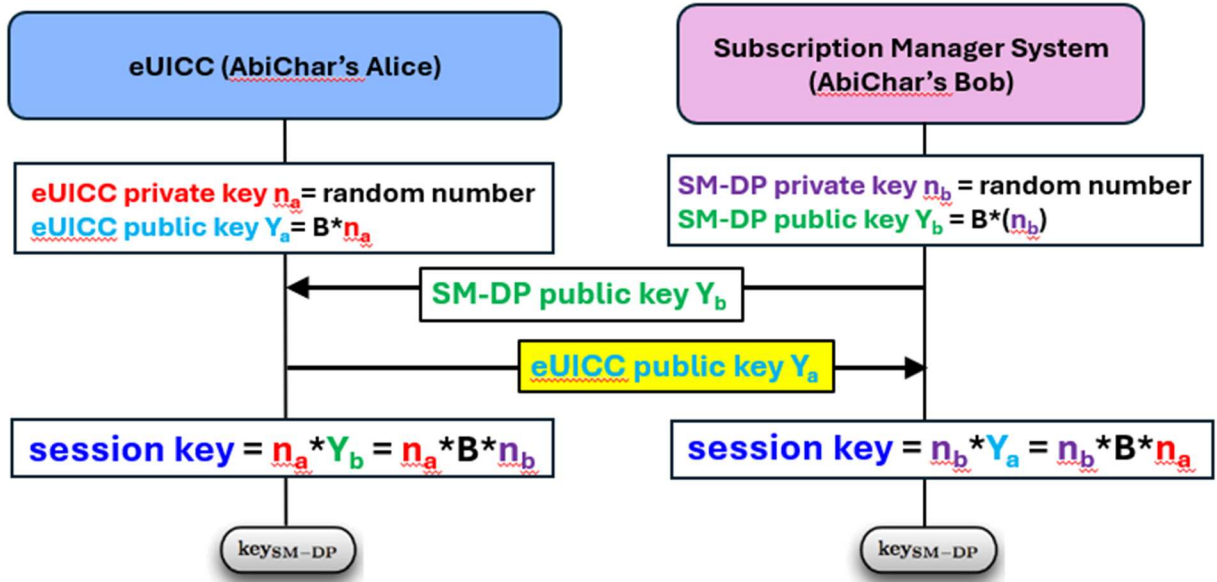
(1) **transmit the eUICC identity to a subscription manager**

145. As explained for Element 1[a], the eUICC transmits its eUICC ID to subscription manager in the initial capability negotiation message, as illustrated in Figure 4, excerpted below. Ex-1005, 300.



(2) **transmit the eUICC public key to the subscription manager**

146. As explained for Element 1[b], in the combined system, the KAM in Park’s eUICC uses AbiChar’s ECDH key agreement protocol to generate a private key n_a and a **corresponding public key Y_a (claimed “eUICC public key”)**. Ex-1007, 3. That public key is then sent to the SM-DP (via SM-SR), as illustrated below, so that it can be combined with a private key n_b generated by the SM-DP, to generate the SM-DP credential/session key, key_{SM-DP} . *Id.* As Park notes, key_{SM-DP} can be generated by asymmetric key agreement (*id.*) of which ECDH is one of the most common choices. For example, *see* Ex-1008, 12.

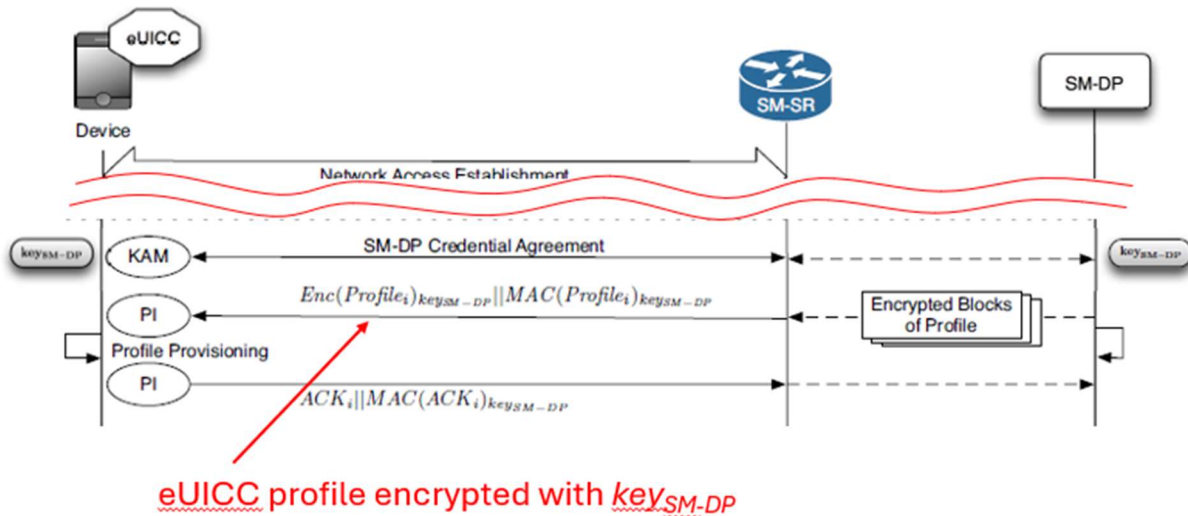


- f. Element 1[c][2]: b. receive, from the subscription manager, i) an eUICC profile comprising network parameters, a key K, and a subscriber identity and ii) a symmetric key; and

147. Park-GlobalPlatform-AbiChar teaches Element 1[c][2].

(1) eUICC profile

148. After performing AbiChar's key agreement to generate the shared SM-DP credential/session key key_{SM-DP} , Park teaches that the SM-DP encrypts a profile "[u]sing the agreed SM-DP Credentials [key_{SM-DP}] and the selected cryptographic algorithms," and then "sends the protected profile blocks to the eUICC." Ex-1005, 300; Fig. 4.



149. Park teaches that the profile sent to the eUICC includes “the MNO’s data and applications ... such as *NAA (Network Access Application)* [] with related data (e.g. *IMSI (International Mobile Subscriber Identity)*, *keys*, authentication algorithm, etc.).” Ex-1005, 297 (describing profiles in the context of traditional UICCs, and explaining that MNOs must be able to remotely provision those same profiles onto eUICCs). A POSITA would have understood that the NAA defines network parameters for enabling an eUICC to attach to, authenticate with, and communicate over the MNO’s network.

(2) a symmetric key

150. Park teaches that the eUICC profile “includes [] security-sensitive information.” Ex-1005, 299. A POSITA would have recognized that such sensitive information includes the keys (claimed “key K”) and IMSI (claimed “subscriber identity”) within the profile. Ex-1005, 297; GlobalPlatform teaches that, when

transmitting sensitive data, such as “secret or private keys,” an additional layer of protection is provided “over and beyond the security level required for the Secure Channel,” by first encrypting the sensitive data with a Data Encryption Key (DEK), and then encrypting it again using the session key, as illustrated below, using GlobalPlatform’s Figure F-11. Ex-1006, 255, 258, 260, 264; Fig. F-11 (annotated). DEK is the recited “symmetric key,” as explained below.

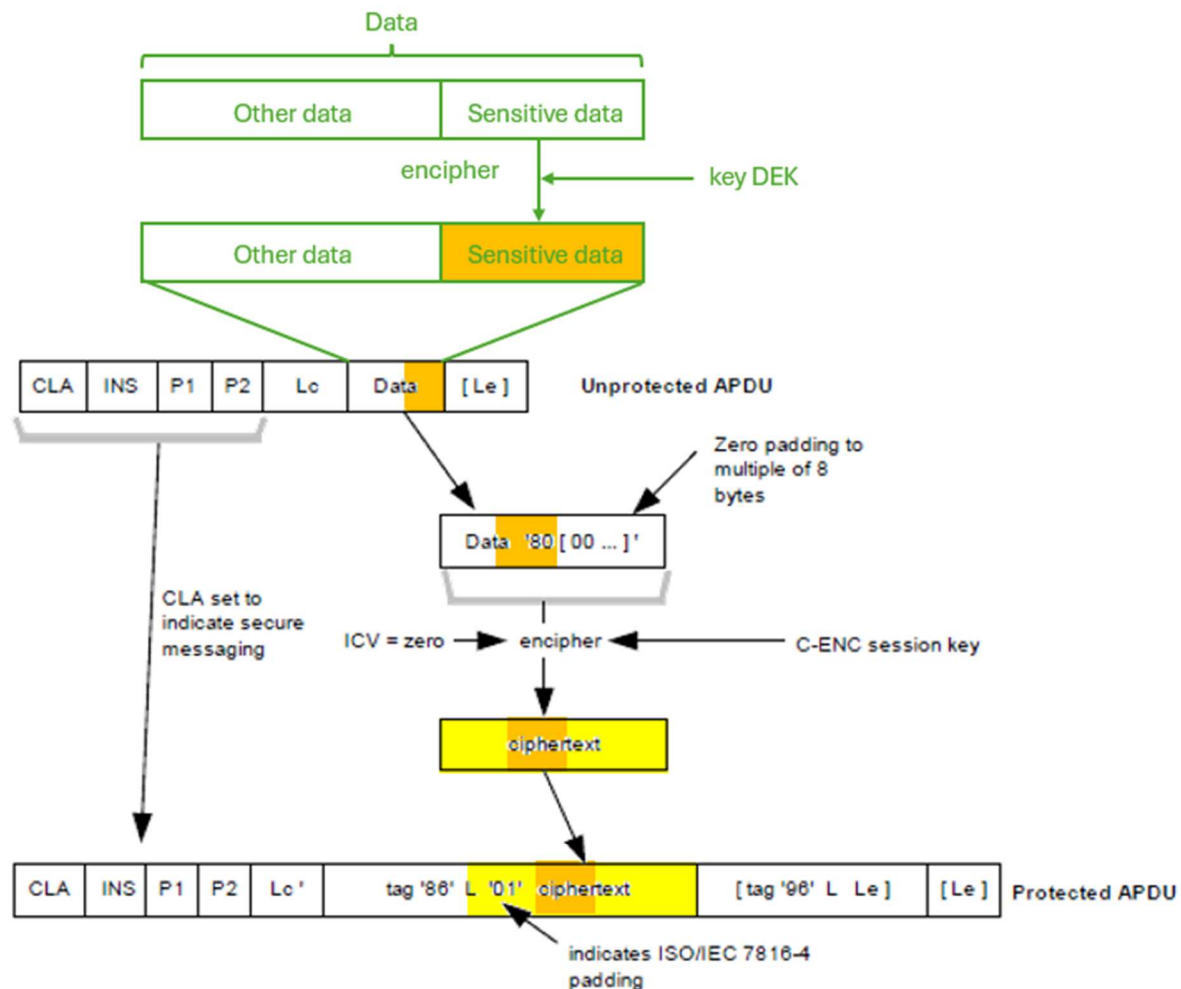


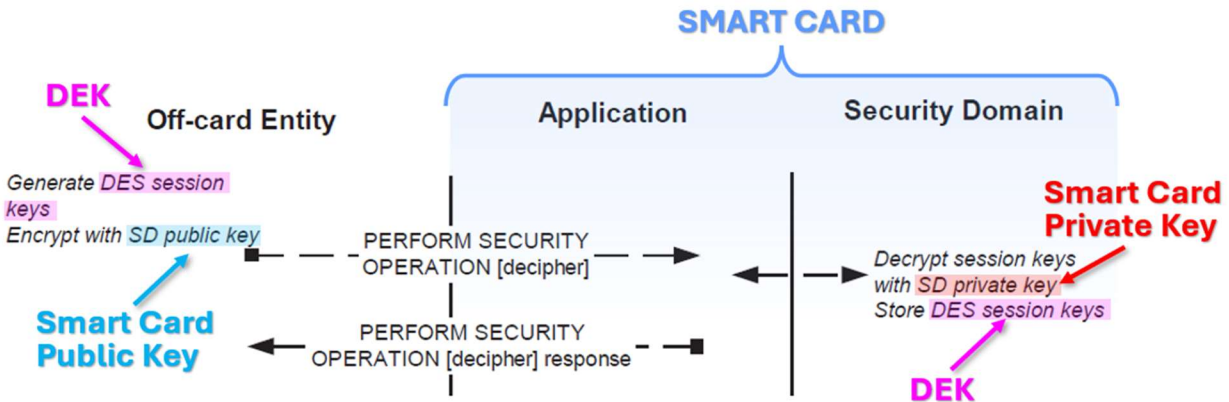
Figure F-11: Secure Messaging: Command message protected for confidentiality

151. A POSITA would have recognized that applying GlobalPlatform's DEK-based inner encryption to the sensitive data included within Park's profile adds an additional layer of protection for the most sensitive data within the profile.

152. GlobalPlatform explains that encryption with the DEK key "uses DES" (Ex-1006, 264), and that "[t]he Data Encryption Standard (DES) is a symmetric cryptographic algorithm that requires the use of the same secret key to encrypt and decrypt data." *Id.*, 174. Thus, in the combined system, GlobalPlatform's symmetric DEK key (claimed "symmetric key") would be used to encrypt the sensitive data (e.g., keys, IMSI) included within Park's profile.

153. Because the DEK is a symmetric key, in order for the eUICC to decrypt a key K encrypted with the DEK, the eUICC must have its own copy of the DEK. GlobalPlatform teaches a "key transport option" that may be used by an off-card entity to send the DEK to a smart card. Ex-1006, 251-52. The DEK is sent to the card using the "PERFORM SECURITY OPERATION [decipher] command." Ex-1006, 255-56 (explaining that the PERFORM SECURITY OPERATION [decipher] command includes "control reference templates" that "contain the actual key values," with "control reference template 'B8'" used to transmit the DEK). GlobalPlatform further teaches that keys sent to smart card using the key transport option are encrypted using the card's public key. Ex-1006, pp.247, 275-76. This is

illustrated in Figure F-9, excerpted below:



Ex-1006, Fig. F-9 (excerpted)

154. A POSITA would have recognized that in the combined Park-GlobalPlatform system, the SM-DP would send the DEK (claimed “symmetric key”) to the eUICC, encrypted with the eUICC’s public key (enabling the eUICC to decrypt it using its corresponding private key). Thus, Park-GlobalPlatform-AbiChar teaches that the eUICC-installed device receives a “symmetric key” from the “subscription manager,” as claimed.

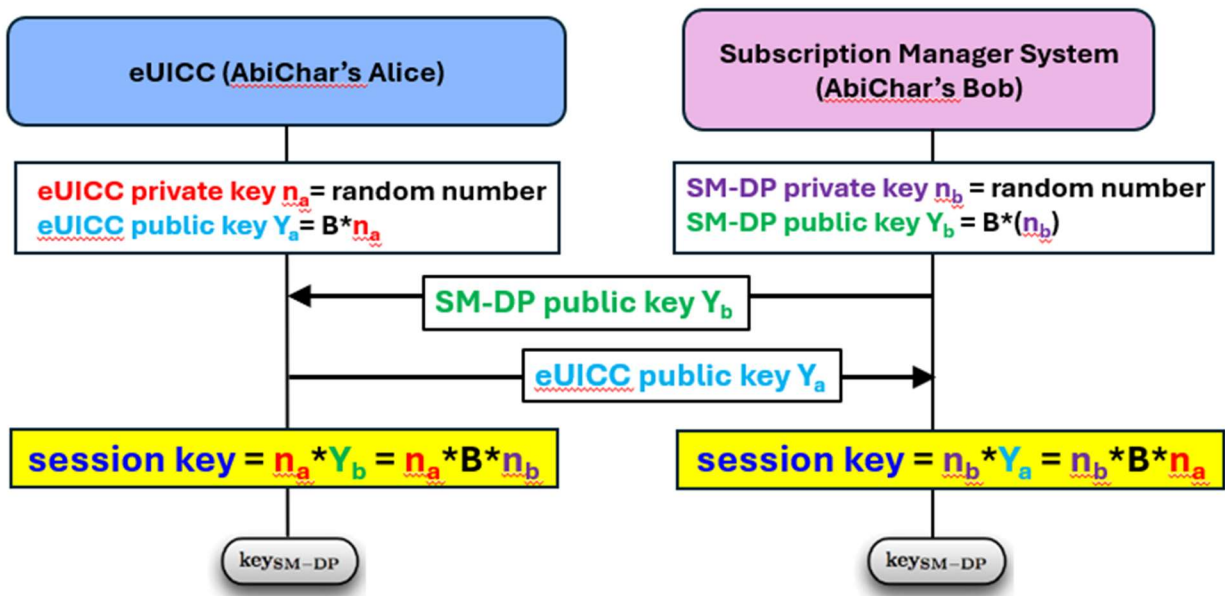
g. Element 1[d]: an eUICC associated with the eUICC identity and configured to

155. Park-GlobalPlatform-AbiChar teaches Element 1[d]. As explained for Element 1[a], Park teaches an eUICC associated with “the eUICC ID (*eICCID*).” Ex-1005, 300.

- h. Element 1[d][1]: a. derive a **profile key** using an elliptic curve Diffie-Hellman (ECDH) key exchange with the **eUICC private key** and a **subscription manager public key**;

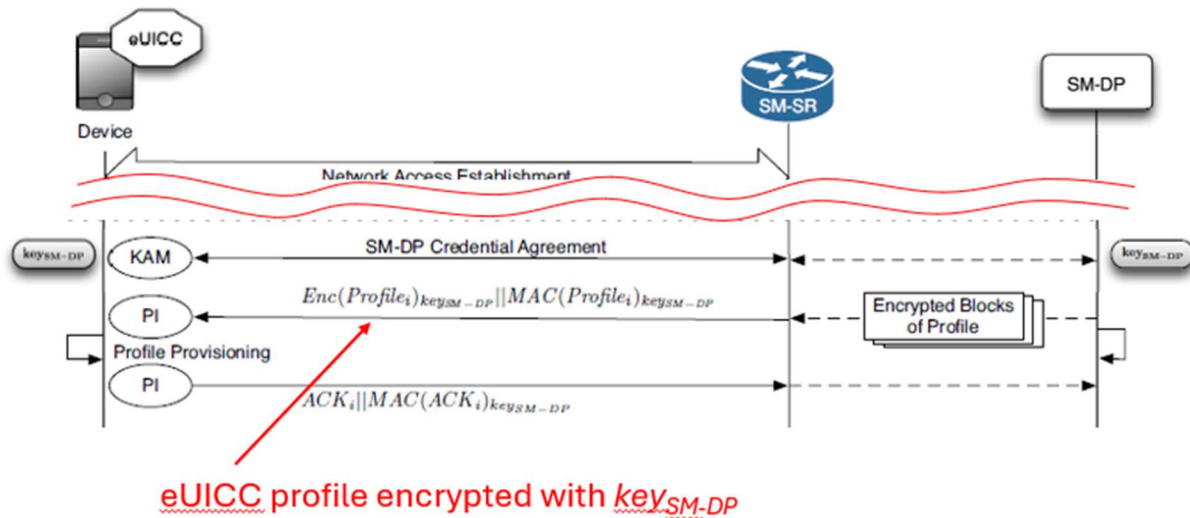
156. Park-GlobalPlatform-AbiChar teaches Element 1[d][1].

157. As explained for Element 1[b], in the combined system, Park's eUICC and SM-DP engage in AbiChar's elliptic curve Diffie-Hellman (ECDH) key exchange to each generate a copy of Park's "SM-DP Credential key_{SM-DP} "/AbiChar's session key, as illustrated below. Ex-1007, 3; Ex-1005, 300.



158. In particular, the eUICC calculates the **shared SM-DP Credential key_{SM-DP}** /session key (claimed "profile key") using its **private key n_a** (claimed "eUICC private key"), together with the **SM-DP public key Y_b** (claimed "subscription manager public key") that it received from the SM-DP. Ex-1007, p.3. Park's key_{SM-DP}

DP results from the KAM (applying AbiChar's ECDH algorithm) and is used to encrypt the eUICC profile as shown below. Ex-1005, Fig. 4 (excerpt).



- i. **Element 1[d][2]: b. decrypt a first portion of the eUICC profile using the profile key;**

159. Park-GlobalPlatform-AbiChar teaches Element 1[d][2].

160. Park teaches that, upon receiving the encrypted profile from the SM-DP, “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.” Ex-1005, 300, Table1 (“Profile Installer (PI) | The software running in the eUICC to perform the decryption and verification of the protected profile blocks from SM-DP”). Because the SM-DP and eUICC both share the same **SM-DP Credential key_{SM-DP} (claimed “profile key”)**, a POSITA would have understood that the SM-DP Credential is a symmetric key—i.e., one that can be used for both

encryption and decryption. Thus, a POSITA would understand that decrypting each profile block, as performed by the eUICC, would involve using the **SM-DP Credential key_{SM-DP} (claimed “profile key”)**.

j. Element 1[d][3]: c. receive the symmetric key from a network application operating in the mobile device;

161. Park-GlobalPlatform-AbiChar teaches Element 1[d][3].

162. As explained for Element 1[c][2], Park-GlobalPlatform-AbiChar teaches that the eUICC-installed device receives GlobalPlatform’s symmetric key DEK (claimed “symmetric key”) from the SM-DP. As illustrated in Figure 5, below, messages sent to the eUICC are received using Park’s Device Module (DM), which is “software running in the Device to perform the roles of the intermediary between eUICC and SM-SR.” Ex-1005, 299. Park’s DM corresponds to the claimed “network application.”

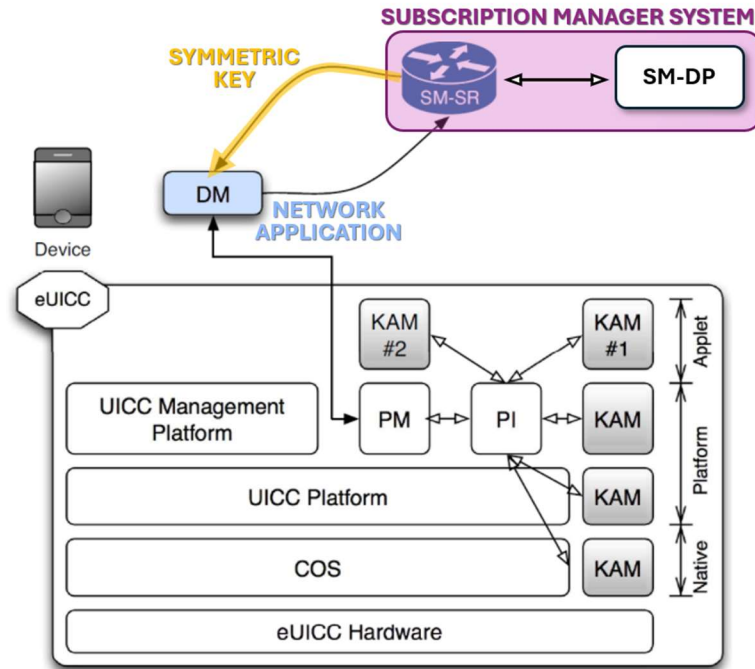


Fig. 5. Internal Architecture of eUICC

- k. **Element 1[d][4]: d. decrypt a second portion of the eUICC profile using the symmetric key, the second portion comprising the key K and the subscriber identity, wherein the first portion and the second portion are distinct; and**

163. Park-GlobalPlatform-AbiChar teaches Element 1[d][4].

164. As explained for Element 1[c][2], in the combined system, GlobalPlatform's DEK (claimed "symmetric key") would be used to encrypt the "security-sensitive information" within the profile (Ex-1005, 299), before the entire profile is wrapped under the SM-DP Credential key_{SM-DP} /session key. A POSITA would have understood that the sensitive information encrypted with the DEK (claimed "symmetric key") would include Park's keys (claimed "key K") and IMSI

(claimed “subscriber identity”), because disclosure of the keys could compromise network authentication and disclosure of the IMSI could allow cloning of the subscriber’s identity.

165. GlobalPlatform teaches that, in response to receiving such encrypted “sensitive data,” a smart card will perform a “decryption” process that “is the exact opposite” of the encryption process. Ex-1006, 264. Thus, in the combined Park-GlobalPlatform system, once the eUICC has decrypted the outer layer using the SM-DP Credential key_{SM-DP} /session key, it would use the DEK to decrypt the inner ciphertext containing the key K and IMSI.

166. Park teaches that the profile also includes a “Network Access Application” (NAA). Ex-1005, 297. As explained for Element 1[c][2], a POSITA would have understood that the NAA includes network parameters used to establish network connectivity. A POSITA would have understood that such network parameters would include non-security-sensitive information—e.g., configuration data shared across all subscribers of the same MNO and/or MNO service. Encrypting and decrypting such information with GlobalPlatform’s DEK would add unnecessary computational overhead to the system. Accordingly, in the combined system, the inner layer of encryption with GlobalPlatform’s DEK (claimed “symmetric key”) would only be applied to a subset of Park’s profile (claimed

“second portion of the eUICC profile”), while the outer layer of encryption with Park’s SM-SR Credentials key_{SM-DP} /AbiChar’s session key (claimed “profile key”) would be applied to the entire profile (claimed “first portion of the eUICC profile”). Thus, Park-GlobalPlatform-AbiChar teaches that “the first portion and the second portion” of the profile “are distinct,” as claimed.

167. Alternatively, once the sensitive data portion (keys, IMSI) has been encrypted using DEK, it is not necessary to further encrypt that portion of the profile with key_{SM-DP} , and it would be a simple matter of design choice (computational complexity vs. cryptographic security) to encrypt the sensitive portion only with DEK and the non-sensitive portion only with key_{SM-DP} since all of the profile would still be protected under at least one key and the sensitive portion would remain accessible only to an entity possessing the claimed “symmetric key” DEK. This would maintain the first and second portions as distinct.

I. Element 1[d][5]: e. generate a response value for authentication of the mobile device with the wireless network using the key K.

168. Park-GlobalPlatform-AbiChar teaches Element 1[d][5].

169. Park teaches that, following successful decryption of the profile, “the profile blocks are provisioned inside the eUICC.” Ex-1005, 300. The provisioned profile includes a “Network Access Application” “with related data,” including

“keys” (claimed “key K”) and an “authentication algorithm.” Ex-1005, 297; A POSITA would have recognized that the key K and authentication algorithm in Park’s profile would be used by the eUICC to authenticate with and gain access to the associated MNO’s network. In particular, a POSITA would have recognized that a well-known method for authentication involves demonstrating possession of a secret key K, by using that key to sign a random challenge issued by the authenticating entity. For instance, GlobalPlatform teaches an authentication method in which the card and off-card entity authenticate one another by proving that they both have “knowledge of the same secret(s),” by generating and verifying cryptographic response values derived from those shared secret(s). Ex-1006, 209-11. And, in the standard 3GPP authentication and key agreement (AKA) process, the eUICC uses its stored secret key K and authentication algorithm (e.g., MILENAGE or COMP128) to compute a response value (RES) to a random challenge sent by the network. Indeed, after describing that a profile includes “keys [and an] authentication algorithm,” Park cites to a technical report on “Reprogrammable SIMs,” which describes this well-known authentication procedure. Ex-1005, 297, 303; Ex-1027, 21-22 (explaining that, as part of the authentication process, a SIM “us[es] the RAND [received from the network] and its own copy of Ki and the A3 algorithm ... to generate the signed response (called

SRES) and passes this to the network”).

170. Thus, Park-GlobalPlatform-AbiChar renders obvious Claim 1.

3. Dependent Claims 3, 6-11, 13-17

- a. Claim 3: The mobile device of claim 1, wherein the eUICC operates within a universal integrated circuit card within the mobile device.**

171. Park-GlobalPlatform-AbiChar teaches Claim 3. Park teaches that the eUICC “is a new form of UICC (Universal Integrated Circuit Card) [], soldered into a device during manufacturing.” Ex-1005, 297.

- b. Claim 6: The mobile device of claim 1, wherein the network application is configured to communicate with i) the wireless network and ii) the eUICC in the mobile device using a system bus.**

172. Park-GlobalPlatform-AbiChar teaches Claim 6.

173. As illustrated in Figure 5, below, Park’s DM (claimed “network application”) communicates with both the SM-SR and the eUICC:

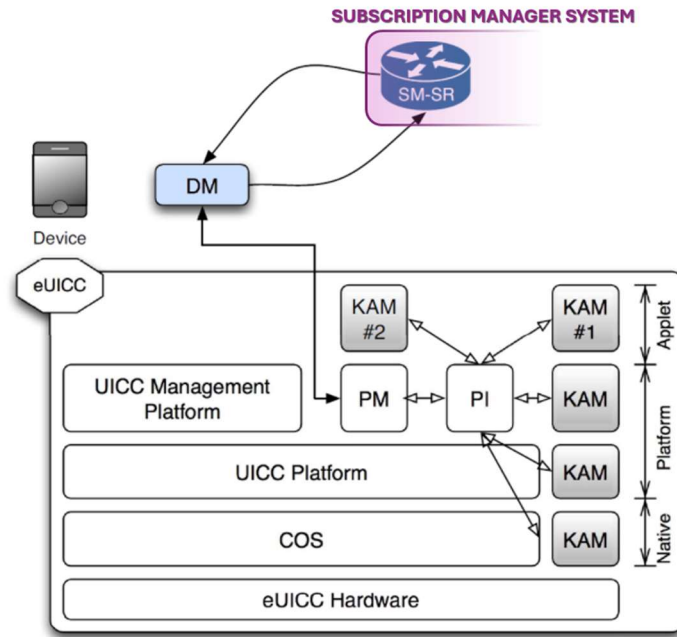


Fig. 5. Internal Architecture of eUICC

174. As explained for Element 1[c], a POSITA would have understood that Park's DM would interface with the device's radio—including its transmit and receive antenna(s)—to enable the wireless communications between the DM and the SM-SR. A POSITA would further have understood that the device processor executing the DM would communicate with the eUICC and the radio over a system bus. Indeed, it was well-known to use a system bus to enable communications between hardware components in a mobile device.

- c. **Claim 7: The mobile device of claim 1, wherein the eUICC comprises computer executable instructions for a processor in the mobile device, and wherein the computer executable instructions are stored within the memory.**

175. Park-GlobalPlatform-AbiChar teaches Claim 7.

176. Park teaches that the eUICC includes “eUICC Hardware,” together with a “software running in the eUICC,” such as the “Key Agreement Module (KAM),” “Profile Installer (PI),” and “Profile Manager (PM).” Ex-1005, 299. A POSITA would have understood that the eUICC’s software modules correspond to machine-readable instructions stored in the eUICC’s memory, that are executed by the eUICC’s processor(s) (claimed “processor in the mobile device”) to perform the described operations, such as key agreement and profile installation.

- d. **Claim 8: The mobile device of claim 1, wherein the eUICC comprises a package soldered onto a circuit board of the mobile device.**

177. Park-GlobalPlatform-AbiChar teaches Claim 8. Park teaches that the eUICC “is a new form of UICC (Universal Integrated Circuit Card) [], soldered into a device during manufacturing.” Ex-1005, 297.

- e. **Claim 9: The mobile device of claim 1, further comprising a user interface configured to receive user identification information before the mobile device receives the symmetric key.**

178. Park-GlobalPlatform-AbiChar teaches Claim 9.

179. GlobalPlatform teaches that a smart card or secure element (such as an eUICC) may be protected by a “Global PIN” that provides “a mechanism for a Cardholder Verification Method,” to prevent “access violations.” Ex-1006, p.71. A POSITA would have understood that this operation involves the eUICC being operably connected to a host module—such as a handset, terminal, or other interface device—that provides a user interface (e.g., keypad or touchscreen) through which the user can enter the Global PIN (claimed “user identification information”).

180. A POSITA would have understood that requiring a user to enter the Global PIN before provisioning the eUICC with secret keys, such as the DEK (claimed “symmetric key”), would provide an additional layer of protection by ensuring that only an authorized user or operator could trigger a “PERFORM SECURITY OPERATION [decipher] command” (Ex-1006, 255-56) to load a new DEK onto the eUICC.

- f. Claim 10: The mobile device of claim 1, wherein the first portion of the eUICC profile includes the network parameters, the network parameters comprising a list of numbers associated with a mobile network operator.**

181. Park-GlobalPlatform-AbiChar teaches Claim 10.

182. As explained for Element 1[d][4], Park teaches that the first portion of the profile includes a “Network Access Application” (NAA) with network parameters associated with the MNO that issued the profile. It would have been well-

known to a POSITA that such parameters would include a list of numbers. For instance, according to well-known 3GPP/ETSI standards, the network parameters would include the “Public Land Mobile Network” (PLMN) list—a list of MNO-preferred networks, in priority order, that the eUICC should connect to when roaming, each identified numerically using the operator’s mobile country code (MCC) and mobile network code (MNC). Indeed, the “Reprogrammable SIMs” technical report cited by Park highlights this PLMN list as being one of the pieces of information stored on a SIM profile. Ex-1027, 17 (“The Public Land Mobile Network (PLMN) selector is a list of MNO-preferred networks in priority order ... [which] allows the MNO to control which visited networks to use (i.e. when roaming or coverage is lost).”).

- g. Claim 11: The mobile device of claim 1, wherein the network parameters comprise a mobile country code (MCC) and a mobile network code (MNC) associated with the wireless network.**

183. Park-GlobalPlatform-AbiChar teaches Claim 11 for the reasons explained for Claim 10.

- h. Claim 13: The mobile device of claim 1, wherein the subscriber identity comprises an International Mobile Subscriber Identity (IMSI).**

184. Park-GlobalPlatform-AbiChar teaches Claim 13. As explained for Element 1[c][2], Park teaches that the profile provisioned to the eUICC includes an

“IMSI (International Mobile Subscriber Identity).” Ex-1005, 297.

- i. **Claim 14: The mobile device of claim 1, wherein the radio is further configured to receive, from the subscription manager, a ciphertext for the symmetric key, wherein the mobile device is configured to decrypt the ciphertext using at least the eUICC private key.**

185. Park-GlobalPlatform-AbiChar teaches Claim 14. As explained for Element 1[c][2], in the combined system, the SM-DP would send the DEK (claimed “symmetric key”) to the eUICC, encrypted with the eUICC’s public key. The encrypted DEK corresponds to the claimed “ciphertext.” A POSITA would have understood that, under standard principles of public-key cryptography, data encrypted with a public key must be decrypted with the corresponding private key. Thus, where the SM-DP encrypts the DEK using the eUICC’s public key, the eUICC necessarily uses its private key to decrypt that ciphertext and recover the DEK. *Id.*

- j. **Claim 15: The mobile device of claim 1, wherein the mobile device further comprises at least one of a wireless handset, a cellular phone, a smartphone, a tablet computer, a laptop, a tracking device, and a circuit board with the radio.**

186. Park-GlobalPlatform-AbiChar teaches Claim 15.

187. Park teaches that the field of use of the eUICC has extended to include “Consumer Electronic Devices.” Ex-1005, p.297. Indeed, as illustrated in Figure 5, below, Park illustrates its eUICC as being housed within a smartphone:

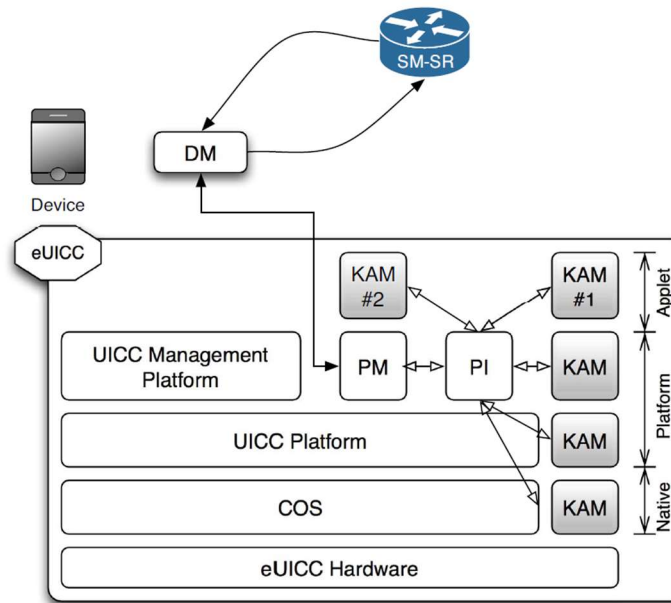


Fig. 5. Internal Architecture of eUICC

188. As explained for Element 1[c], A POSITA would have understood that Park's DM would interface with a radio on the device to enable wireless communications with the SM-SR. A POSITA would have recognized that such a radio would be included in a circuit board of the device.

k. Claim 16: The mobile device of claim 1, wherein the second memory is the same memory as the first memory.

189. Park-GlobalPlatform-AbiChar teaches Claim 16.

190. A POSITA would have understood that the "first memory" for storing an eUICC identity (which would likely be nonvolatile) could also be the "second memory" used in conjunction with the random number generator, especially if the

random eUICC private key n_a needs to be available through a power cycle. A POSITA would have viewed this as being a simple matter of design choice, since the same secure memory can efficiently hold both identification and cryptographic data without impacting functionality or security.

I. Claim 17: The mobile device of claim 1, wherein the one or more receiving antennas are the one or more transmit antennas.

191. Park-GlobalPlatform-AbiChar teaches Claim 17. It would have been well-known to a POSITA that the same antenna in a mobile device is commonly used for both transmitting and receiving signals over a wireless network. Indeed, the “Reprogrammable SIMs” technical report cited by Park describes a handset with a single antenna. Ex-1027, 15 (explaining that “the SIM [] securely communicate[s] (via the handset and *its antenna*) with the MNO’s transmitters and servers”).

B. Ground 2: Claims 1, 3, 6-11, 13-17 are obvious over Park in view of GlobalPlatform and X9.63-Overview

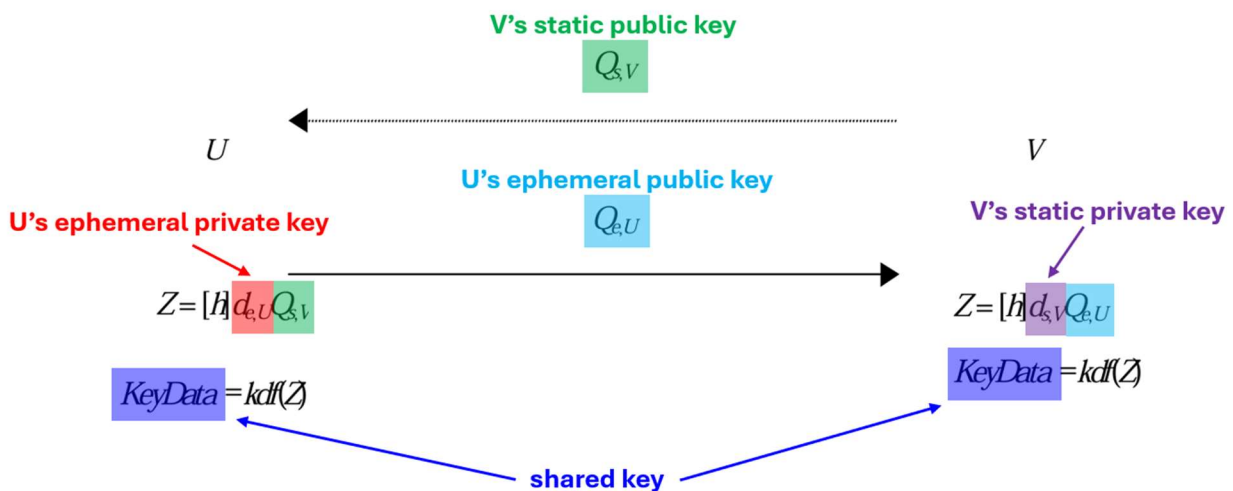
1. A POSITA would have been motivated to combine Park’s teachings with GlobalPlatform’s and X9.63-Overview’s teachings and would have had a reasonable expectation of success.

192. A POSITA would have been motivated to combine Park’s secure profile provisioning architecture (SPA) with GlobalPlatform’s procedures for sensitive data encryption, for the same reasons as explained for Ground 1, above. A POSITA would further have been motivated to combine Park’s SPA with X9.63-

Overview's 1-pass Diffie-Hellman key agreement protocol, to implement a specific embodiment of Park's asymmetric key-exchange mechanism that is well-suited for Park's architectural environment, in which eUICC's are considered "resource-constrain[ed]" devices, for which "overhead for the security-related operations" is undesirable." Ex-1005, p.299.

193. As X9.63-Overview illustrates, and as described above in Section XI.D, the parties exchange public keys, $Q_{s,v}$ and $Q_{e,u}$, and then each side combines the other side's public key with its own private key—for example, the eUICC uses its **random private key** $d_{e,u}$ together with the **received public key** $Q_{s,v}$ —to generate a shared secret:

ANSI X9.63 - 1-Pass DH



Ex-1008, 12. From that shared secret, each party can then derive the same symmetric key, **KeyData**, corresponding to the profile key in claim 1(e), as described below in Section XII.B.2.d. A POSITA would have understood that a one-pass DH exchange offers forward secrecy to the eUICC side of the communication. In the one-pass approach, the eUICC generates a fresh ephemeral public/private key pair for each session, ensuring that even if its long-term private key were later compromised, previously provisioned profiles could not be decrypted. While the same does not necessarily hold true for the server in a one-pass scheme, a POSITA would have recognized that this tradeoff may be acceptable in a profile-provisioning environment such as Park's, where the SM-DM likely resides in a secure infrastructure.

2. Independent Claim 1

a. Elements 1[pre]-1[a], 1[c], 1[c][2]-1[d], 1[d][2]-1[d][5]:

194. Park in view of GlobalPlatform and X9.63-Overview ("Park-GlobalPlatform-X9.63-Overview") teaches Elements 1[pre]-1[a], 1[c], 1[c][2]-1[d], and 1[d][2]-1[d][5] for the same reasons as explained for Ground 1.

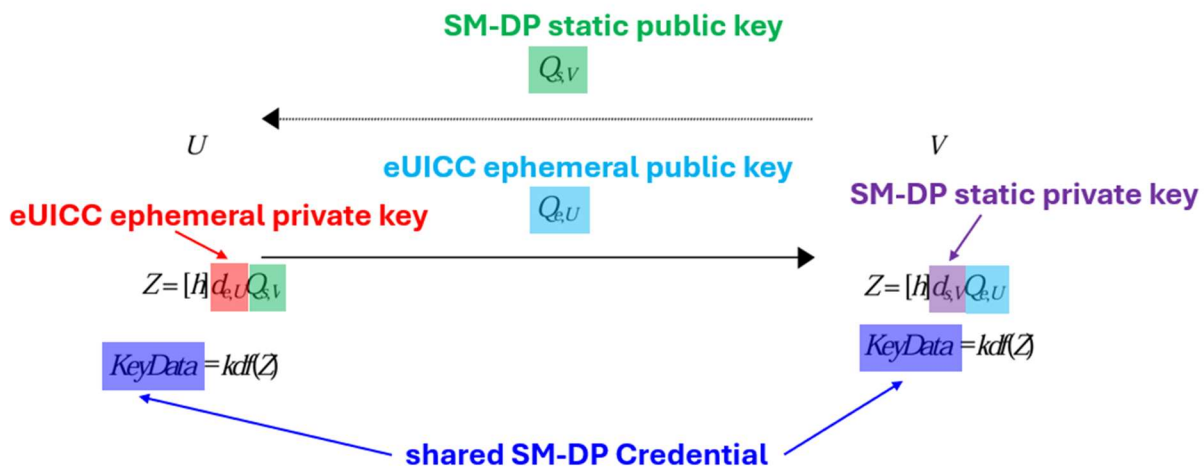
b. Element 1[b]: a random number generator operably connected to a processor connected to a second memory configured to generate a random number for an eUICC private key corresponding to an eUICC public key;

195. Park-GlobalPlatform-AbiChar teaches Element 1[b].

- (1) generating “a random number for an **eUICC private key** corresponding to an **eUICC public key**” using a “random number generator”

196. As explained in Section XII.B.1, a POSITA would have been motivated to use X9.63-Overview’s 1-pass Diffie-Hellman key exchange to implement Park’s “asymmetric algorithm-based key agreement protocol,” as illustrated below:

ANSI X9.63 - 1-Pass DH



Ex-1008, 12. As illustrated, in the combined Park-GlobalPlatform-X9.63-Overview system, the eUICC generates an **ephemeral private key $d_{e,U}$** (claimed “**eUICC private key**”). *Id.* The eUICC then calculates a **corresponding ephemeral public key $Q_{e,U}$** (claimed “**eUICC public key**”). Ex-1008, 12.

197. X9.63-Overview further teaches that **ephemeral private key $d_{e,U}$** is “selected randomly.” *Id.*, 7. A POSITA would have understood that the eUICC

would include a random number generator—e.g., within, or coupled to, Park’s KAM software—for use in generating the random number for **ephemeral private key $d_{e,U}$** .

(2) the random number generator “operably connected to a processor connected to a second memory”

198. Park teaches that the random number generator is operably connected to a processor connected to a second memory, for the same reasons explained for Ground 1.

c. Element 1[c][1]: a. transmit, to a subscription manager, the eUICC identity and the **eUICC public key; and**

199. Park-GlobalPlatform-X9.63-Overview teaches Element 1[c][1].

(1) transmit the eUICC identity to a subscription manager

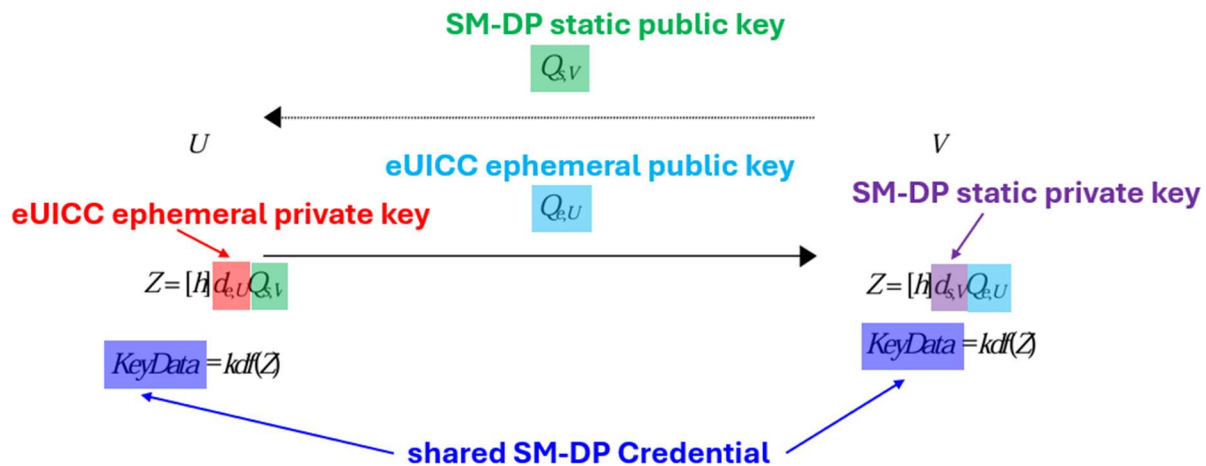
200. Park teaches transmitting the eUICC identity to a subscription manager, for the same reasons explained for Ground 1.

(2) transmit the **eUICC public key to the subscription manager**

201. As explained for Element 1[b], in the combined system, the KAM in Park’s eUICC uses X9.63-Overview’s 1-pass Diffie-Hellman key exchange to generate a private key $d_{e,U}$ and **corresponding public key $Q_{e,U}$ (claimed “eUICC public key”)**. Ex-1008, 12. That public key is then sent to the SM-DP (via SM-SR),

as illustrated below, so that it can be combined with a private key $d_{s,v}$ generated by the SM-DP, to generate the SM-DP credential/session key. *Id.*

ANSI X9.63 - 1-Pass DH

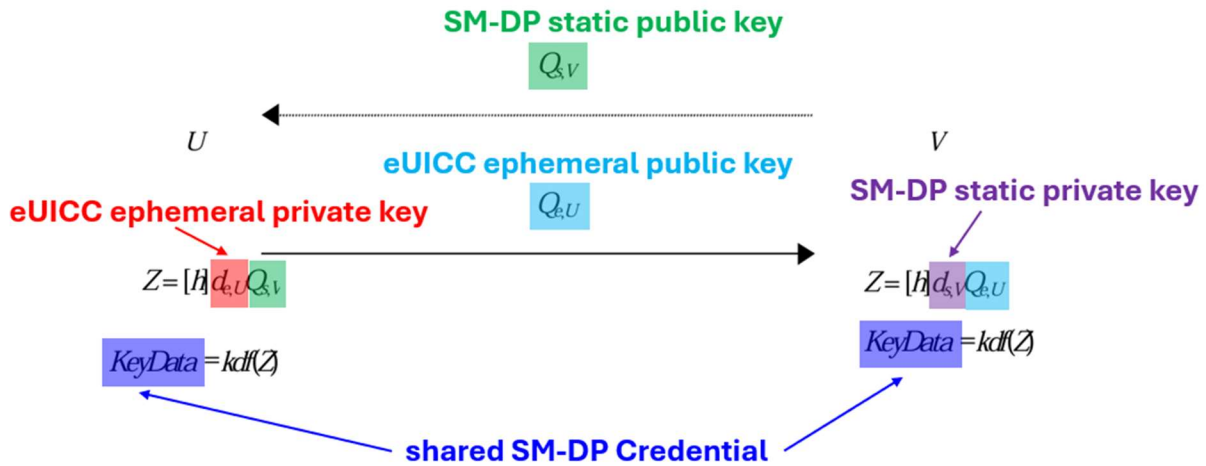


- d. Element 1[d][1]: a. derive a **profile key** using an elliptic curve Diffie-Hellman (ECDH) key exchange with the **eUICC private key** and a **subscription manager public key**;

202. Park-GlobalPlatform-X9.63-Overview teaches Element 1[d][1].

203. As explained for Element 1[b], in the combined system, Park's eUICC and SM-DP engage in X9.63-Overview's 1-pass Diffie-Hellman key exchange to each generate a copy of Park's "SM-DP Credential," as illustrated below. Ex-1008, 12; Ex-1005, 300.

ANSI X9.63 - 1-Pass DH



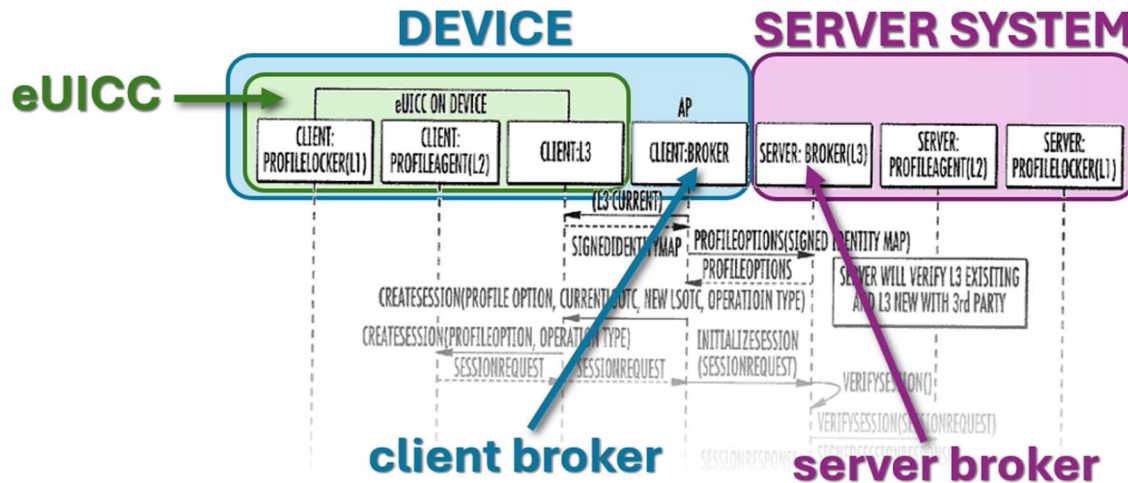
3. Dependent Claims 3, 6-11, 13-17

204. Park-GlobalPlatform-X9.63-Overview teaches Claims 3, 6-11, and 13-17 for the same reasons as explained for Ground 1.

C. Ground 3: Claims 2 and 12 are obvious over Grounds 1 or 2 in view of Haggerty

1. A POSITA would have been motivated to add Haggerty to Park-GlobalPlatform-AbiChar or Park-GlobalPlatform-X9.63-Overview and would have had a reasonable expectation of success.

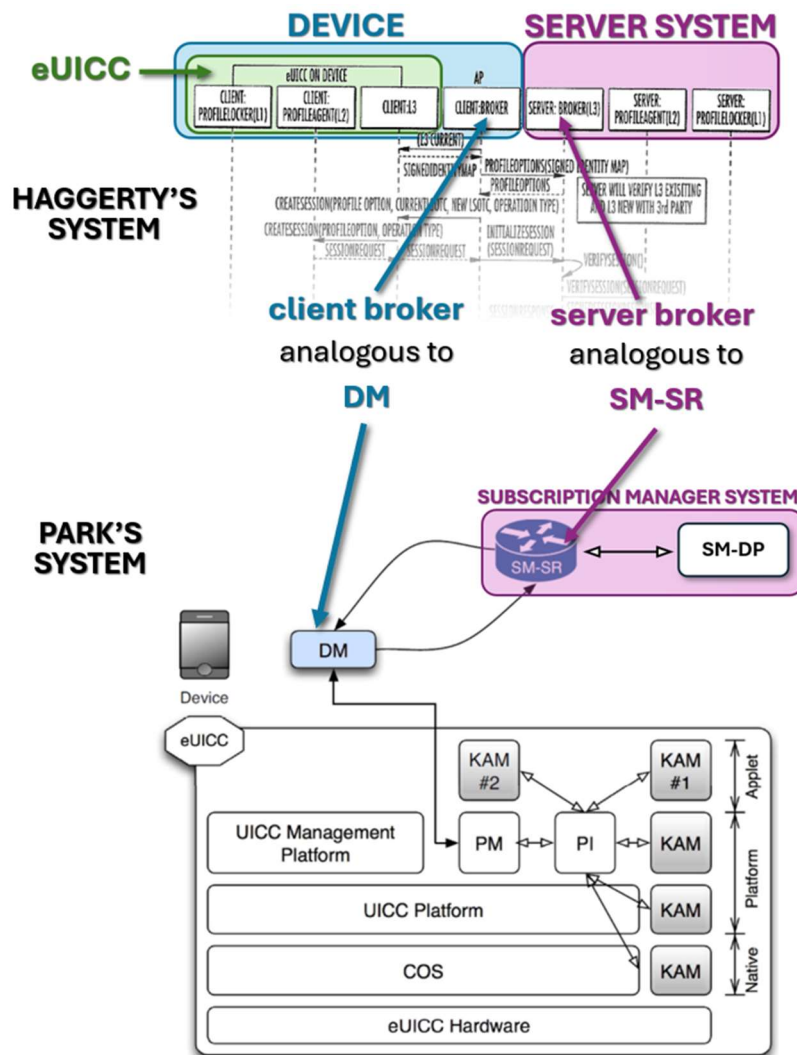
205. Haggerty teaches a broker-based architecture for secure eSIM provisioning in which communications between an eUICC and a backend provisioning server are mediated by two intermediary components—a device broker and a server broker, as illustrated in excerpted Figure 9, below.



206. Haggerty’s device broker resides on the eUICC-containing device, and is responsible for passing eUICC-generated messages, such as session requests or profile download requests, to the provisioning infrastructure. Ex-1013, ¶¶97, 104. The server broker, in turn, operates within the provisioning server system, and is responsible for receiving those messages and forwarding them to backend components such as the “profile agent” or “profile locker,” which handle cryptographic verification and personalization of the eSIM profile. Ex-1013 ¶¶105, 108, Fig. 9. In Haggerty’s system, communications between the two brokers are protected using a standard secure transport layer protocol, such as Transport Layer Security (TLS), to prevent interception or replay attacks. Ex-1013 ¶100.

207. A POSITA would have recognized that Haggerty’s architecture closely parallels the framework disclosed in Park. As illustrated in the figures below,

Haggerty's device broker is analogous to Park's Device Module (DM)—each serves as the intermediary on the device side, interfacing between the eUICC and the remote provisioning infrastructure. Likewise, Haggerty's server broker is analogous to Park's SM-SR—each is responsible for receiving provisioning requests from the device, and routing them to the appropriate profile provisioning components.



208. Similar to Haggerty, Park teaches that the DM and SM-SR

communicate over a “secure communication channel.” Ex-1005, 300. However, Park does not specify what protocol is used to establish or protect that channel, leaving its implementation up to the system designer. *Id.* Haggerty fills that gap by teaching the use of TLS as the underlying secure transport layer. Ex-1013 ¶100. A POSITA would have found it straightforward to incorporate Haggerty’s teaching into Park’s system by using TLS for the secure communication channel between the DM and the SM-SR. A POSITA would have recognized that TLS was a well-established, standardized protocol providing confidentiality, integrity, and mutual authentication—exactly the security objectives Park seeks to achieve.

209. Haggerty additionally teaches that, when provisioning a profile, certain information, such as the ICCID, is kept in plaintext (Ex-1013, ¶84), enabling the provisioning servers to identify and manage the profile without decrypting its full eSIM payload. Ex-1013 ¶¶54, 105. A POSITA would have understood that incorporating Haggerty’s teaching into Park’s system would enable Park’s SM-SR (analogous to Haggerty’s server broker) to use the plaintext ICCID for profile identification and routing, while maintaining end-to-end encryption for the sensitive profile content. This would improve operational efficiency and scalability—allowing Park’s SM-SR to handle multiple profiles—without compromising confidentiality.

2. Claim 2: The mobile device of claim 1, wherein the eUICC is further configured to receive a profile identity for the eUICC profile as a plaintext.

210. Park-GlobalPlatform-AbiChar-Haggerty and Park-GlobalPlatform-X9.63-Overview-Haggerty teach Claim 2.

211. Haggerty teaches that, when encrypting an eSIM for delivery to an eUICC, certain metadata fields, including the Integrated Circuit Card Identifier (“ICCID”), should remain as plaintext. Ex-1012, ¶84. For example, Haggerty’s batch descriptor contains “information for the eSIM” and includes “a plain text section ... that consists of: an ICCID for each profile.” Id. ¶85, Fig. 7. In other words, the plaintext ICCID identifies each profile. In the combined systems, this would allow Park’s SM-SRs and receiving devices to identify profiles without decrypting the full payload. Ex-1012, ¶¶54, 105;

3. Claim 12: The mobile device of claim 1, wherein the radio is further configured to receive, from the subscription manager, the eUICC profile using transport layer security.

212. Park-GlobalPlatform-AbiChar-Haggerty and Park-GlobalPlatform-X9.63-Overview-Haggerty teach Claim 12.

213. Haggerty teaches that “a security protocol (e.g., transport layer security (TLS))” is used to protect communications between Haggerty’s “device broker” (analogous to Park’s DM) and “server broker” (analogous to Park’s SM-SR, respectively). Ex-1013, ¶100. In the combined systems, Haggerty’s TLS would be

used to protect the profile delivered by the SM-SR to the device, to “prevent capture and replay attacks.” Ex-1013, ¶100.

D. Ground 4: Claims 4-5 are obvious over Grounds 1 or 2 in view of Pierce

- 1. A POSITA would have been motivated to add Pierce’s to Park-GlobalPlatform-AbiChar or Park-GlobalPlatform-X9.63-Overview and would have had a reasonable expectation of success.**

214. Both AbiChar and X9.63-Overview teach the use of a random number generator (RNG) to generate private keys for use in an elliptic curve Diffie-Hellman key exchange. Ex-1007, 3; Ex-1008, 7, 12. However, neither reference provides specific details about the RNG.

215. Pierce teaches a method “for use by an embedded processing device within a manufactured product,” to generate strong cryptographic keys using a “pseudo random number generator” (PRNG). Ex-1013, Abstract. Pierce teaches seeding a PRNG with entropy data from sources that are naturally available to such embedded devices, such as sensor outputs, timing data (e.g., GPS time), unique identifiers, and/or transient events on the communication bus, to generate seed values for the PRNG. Ex-1013 ¶¶35-38, Fig. 3. This approach addresses the well-known challenge of generating high-quality randomness in embedded or constrained devices, such as those described in Park, which may lack dedicated “entropy hardware or software engines.” Ex-1013 ¶3; A POSITA would have recognized

that incorporating Pierce's method would improve the security of the Park-GlobalPlatform-AbiChar and Park-GlobalPlatform-X9.63-Overview systems, by helping to ensure that the private keys used in their ECDH exchanges were derived from sufficiently random and unpredictable sources that were readily available to the eUICC-storing devices in these systems.

2. **Claim 4: The mobile device of claim 1, wherein the random number generator is configured to generate the random number in response to input from at least one of a clock and a sensor.**

Claim 5: The mobile device of claim 1, wherein the random number generator is configured to generate the random number from a seed value, wherein the seed value comprises data from at least one of a sensor, the radio, a bus, a clock, a physical interface, the memory, and an operating system.

216. Park-GlobalPlatform-AbiChar-Pierce and Park-GlobalPlatform-X9.63-Overview-Pierce teach Claims 4-5.

217. Pierce teaches a random number generator configured to generate random numbers in response to receiving a seed value in the form of "entropy data" "from a sensor." Ex-1013 ¶36.

E. Ground 5: Claims 10-11 are obvious over Grounds 1 or 2 in view of Konstantinou

1. **A POSITA would have been motivated to add Konstantinou to Park-GlobalPlatform-AbiChar or Park-GlobalPlatform-X9.63-Overview and would have had a reasonable expectation of success.**

218. Park teaches procedures for remotely provisioning profiles onto eUICCs. Ex-1005, 297. However, it does not provide significant detail regarding the content of those profiles. For instance, while Park teaches that profiles include NAAs (Ex-1005, 297), it does not specify what those network access parameters include. Konstantinou fills that gap by teaching that according to “TIA/EIA/IS-683-D: Over-the-Air Service Provisioning of Mobile Stations,” a profile’s MSPL (Multi-mode System Priority List) includes “MCC [mobile country code] / MNC [Mobile Network Code]” entries. Ex-1014 ¶¶31, 158, 163, 166. A POSITA would have turned to Konstantinou’s detailed disclosure to determine what Park’s and SGP.22’s network access parameters include.

- 2. Claim 10: The mobile device of claim 1, wherein the first portion of the eUICC profile includes the network parameters, the network parameters comprising a list of numbers associated with a mobile network operator.**

Claim 11: The mobile device of claim 1, wherein the network parameters comprise a mobile country code (MCC) and a mobile network code (MNC) associated with the wireless network.

219. Park-GlobalPlatform-AbiChar-Konstantinou and Park-GlobalPlatform-X9.63-Overview-Konstantinou teach Claims 10-11.

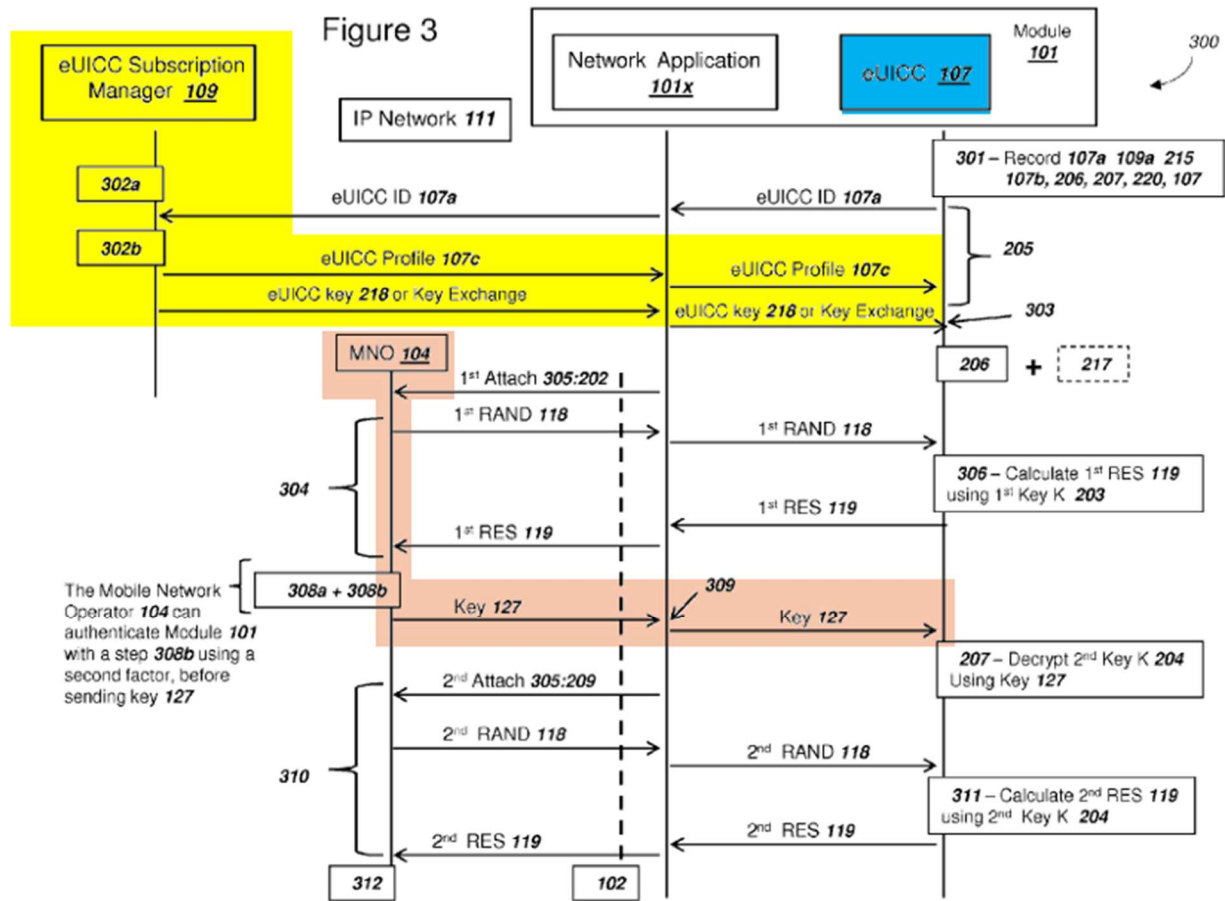
220. As explained in Section YY (Ground 1, Element 1[d][4]), Park teaches that the first portion of the profile includes a Network Access Application (NAA) with network parameters. Konstantinou teaches that such network

parameters include numerical “MCC/MNC” values according to TIA standards. Ex-1014 ¶31. A POSITA would have understood that the MCC and MNC values are numerical.

F. Ground 6: Claims 1-17 are obvious over Nix175 in view of Park and GlobalPlatform

1. A POSITA would have been motivated to combine Nix175, Park, and GlobalPlatform and would have had a reasonable expectation of success.

221. Nix175 discloses the claims of the '893 Patent with the exception of claim element 1[c][2] “receiving, from the subscription manager ... the symmetric key.” As discussed above, the '893 Patent (and Nix175) introduces the claimed “symmetric key” to address a security concern that arises if distribution and release of the profile (including decryption of the secret key K) were to be under the control of the SM and not the MNO. Ex-1016, 3:32-36, 34:27-45. Nix175’s solution to the security problem allows the SM to handle delivery the encrypted profile but requires that the MNO handle delivery of the symmetric key, without which the eUICC cannot decrypt the portion of the profile containing key K. *Id.* 34:54-60. This architecture is depicted in Figure 3 of Nix175, annotated below:



222. Above, the subscription manager (yellow) provides the encrypted eUICC profile 107c to the eUICC (blue). But the eUICC cannot decode key K (“2nd Key K 204) until it receives symmetric key 127 (red) from the MNO 104. Ex-1016,

Fig. 3.

223. While Nix175's introduction of a symmetric key controlled and delivered by the MNO addresses the security issue by preventing the SM from being able to access the secure key K, a POSITA would have recognized that it imposes an undue burden on the MNO, which now must handle key management and

delivery for every eUICC. Thus a POSITA would have been motivated to look to Park, which teaches a split architecture for the subscription manager that divides it into two parts: “SM-DP (Data Preparation)” for generating and encrypting the operator profile, and “SM-SR (Secure Routing)” for transporting and installing the encrypted profile onto the eUICC, as shown in Park’s Figure 1 below. Ex-1005, 297.

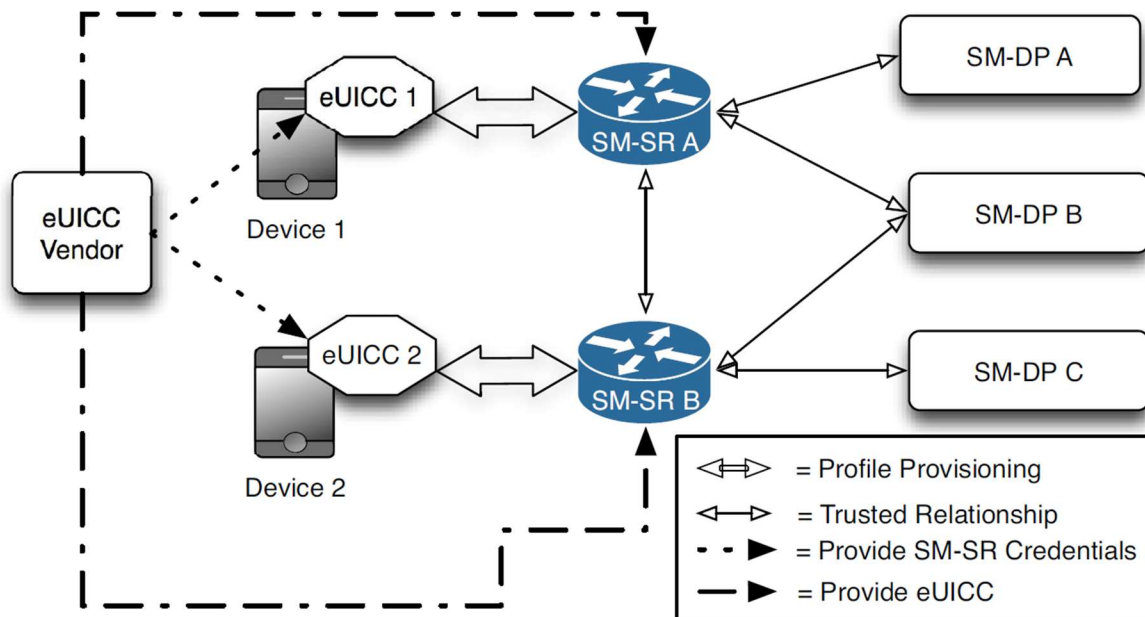


Fig. 1. eUICC Profile Ecosystem

224. By splitting the SM, Park allows for SM-SR, which securely routes the encrypted profile to the eUICC, to exist in a different trust zone³ and potentially be

³ “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.” Ex-1005, 297.

operated by a different entity than SM-DP and the MNO, thereby allowing SM-DP to negotiate profile encryption keys with the eUICC without exposing them to the SM-SR or other intermediaries. Ex-1005, 299-301. Thus, a POSITA would have understood that Park's SM-DP would maintain control of the encryption of critical data like key K, and SM-SR would deliver the encrypted profile to the eUICC but would not have access to the symmetric key, thereby solving the security problem while freeing the MNO from having to directly manage key delivery to each eUICC.

225. In this combined system, a POSITA also would have recognized advantages to continuing to encrypt key K with a symmetric key that is separate from the key used to encrypt the overall profile. In particular, in embodiments of Nix175's system in which the profile key (the symmetric key used to encrypt the overall profile) is generated dynamically through a key-exchange protocol with the eUICC, the SM-DP would not be able to pre-encrypt the profile with the profile key. At the same time, a POSITA would have understood that sensitive data, such as key K, should not be stored at the SM-DP in plaintext. *See, e.g.*, Ex-1023, NIST SP 800-57, 73. Thus, encrypting key K under a separate symmetric key would help ensure that the SM-DP could store and handle key K securely, before establishing the profile key with the eUICC.

226. A POSITA would also have appreciated that retaining key K in its

already-encrypted form, once the profile key has been established, and simply encrypting the entire profile (including already-encrypted key K) with that profile key, would provide additional security benefits, by providing greater protection against compromise during transport. *See, e.g.*, Ex-1024, NIST SP 800-38F, 2. Indeed, Nix175 expressly contemplates this kind of layered encryption. Ex-1016, 33:60-63 (“ciphertext 208b could optionally be included within ciphertext 208a”), 37:57-67 (“a profile 107c can include multiple layers of ciphering”).

227. This same principle is taught by GlobalPlatform, which teaches first encrypting sensitive data within a message “us[ing] the relevant data encryption session key (DEK)” and then further encrypting the message with the secure channel session key, providing layered protection for these highly sensitive materials. Ex-1006, 264, 258 (explaining that the session key, C-ENC, is used to encrypt “the command data field of the command message being transmitted to the card,” which “includes any data within the data field that has *already been protected* for another purpose *e.g. secret or private keys encrypted with the data encryption session key*”). This configuration is illustrated below in annotated Figure F-11.

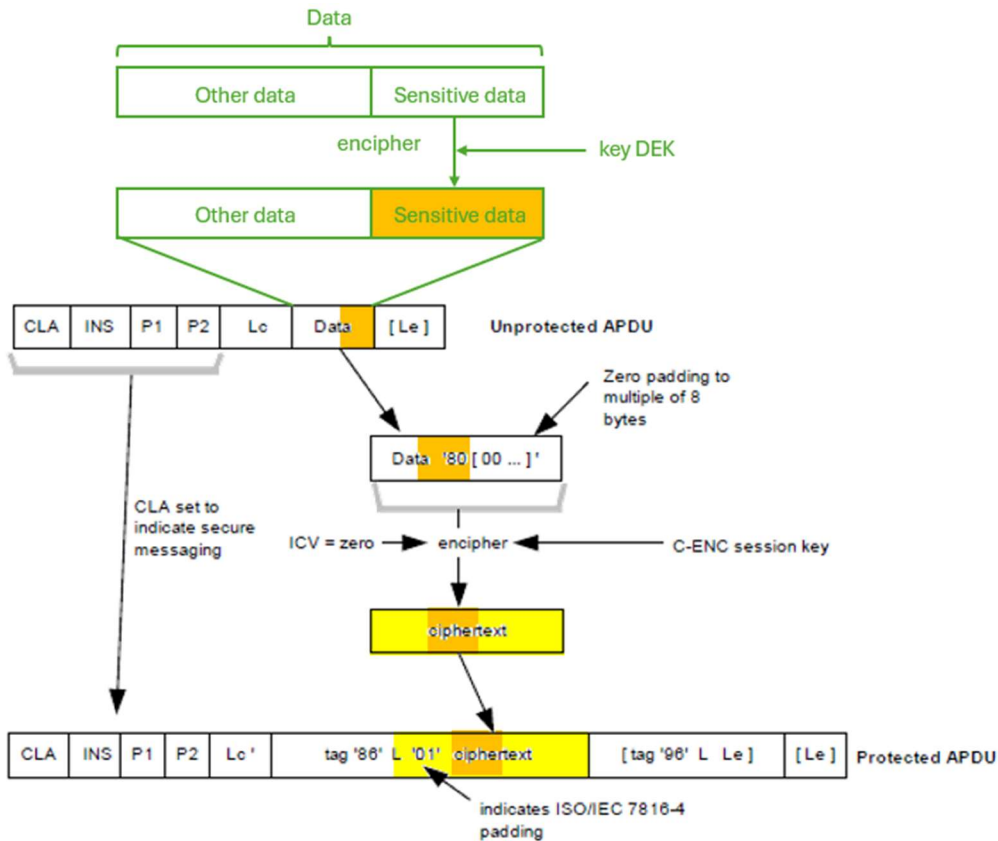


Figure F-11: Secure Messaging: Command message protected for confidentiality

228. Thus, in the combined system, profiles sent from the SM-DP to the eUICC would include an inner layer of encryption for sensitive materials like keys (using GlobalPlatform's DEK), as well as an outer layer of encryption using Nix175's profile key.

229. Alternatively, once the sensitive data portion (keys, IMSI) had been encrypted using DEK, it would not be necessary to further encrypt that portion of the profile with the profile key, and it would be a simple matter of design choice (computational complexity vs. cryptographic security) to encrypt the sensitive

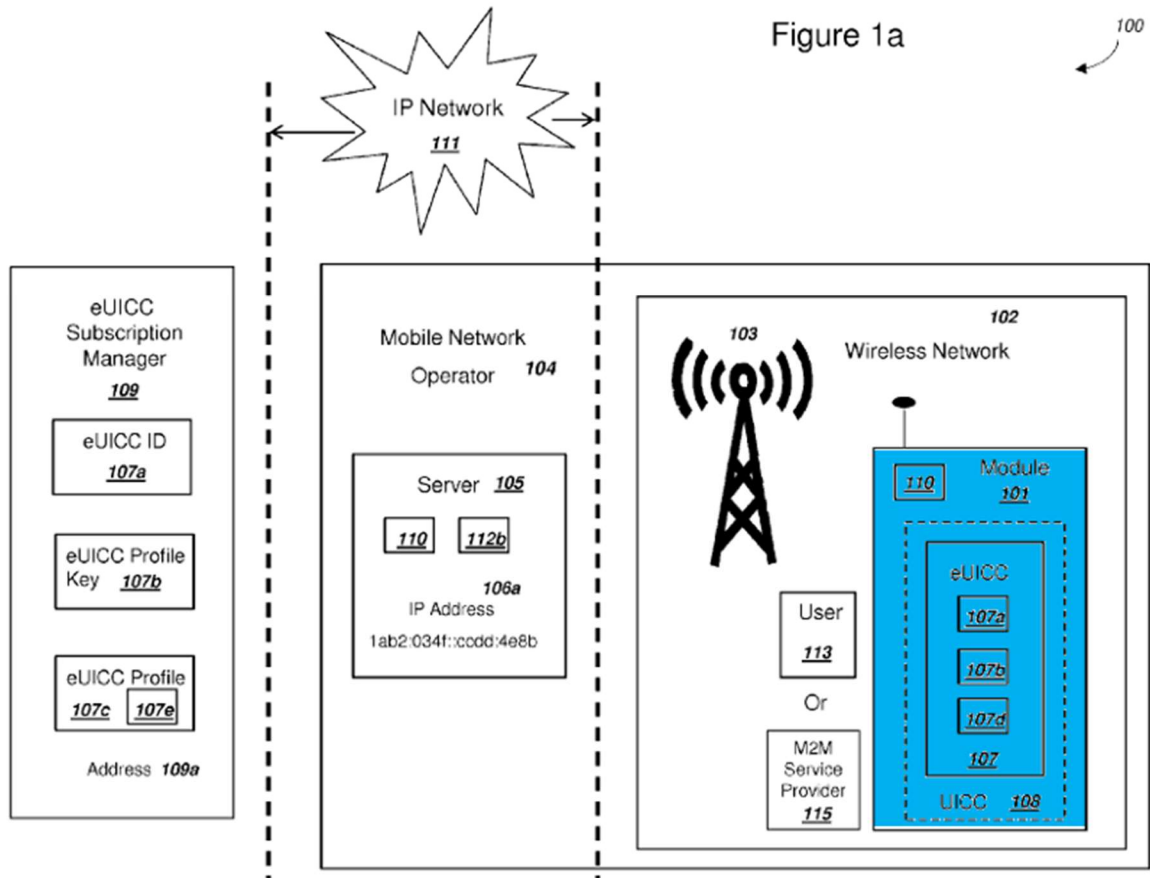
portion only with DEK and the non-sensitive portion only with the profile key, since all of the profile would still be protected under at least one key and the sensitive portion would remain accessible only to an entity possessing the claimed “symmetric key” DEK.

230. Thus, a POSITA would have been motivated to combine Nix175 with Park-GlobalPlatform and would have expected the combination to be successful.

2. Independent Claim 1

a. 1[pre]: A mobile device for communicating with a wireless network, the mobile device comprising:

231. Nix175 teaches “A module in the present invention can comprise a *mobile phone* such as a smartphone, and may also be referred to as a *mobile device*, mobile station, or user equipment.” Ex-1016, 4:24-27; Fig. 1a, module 101.



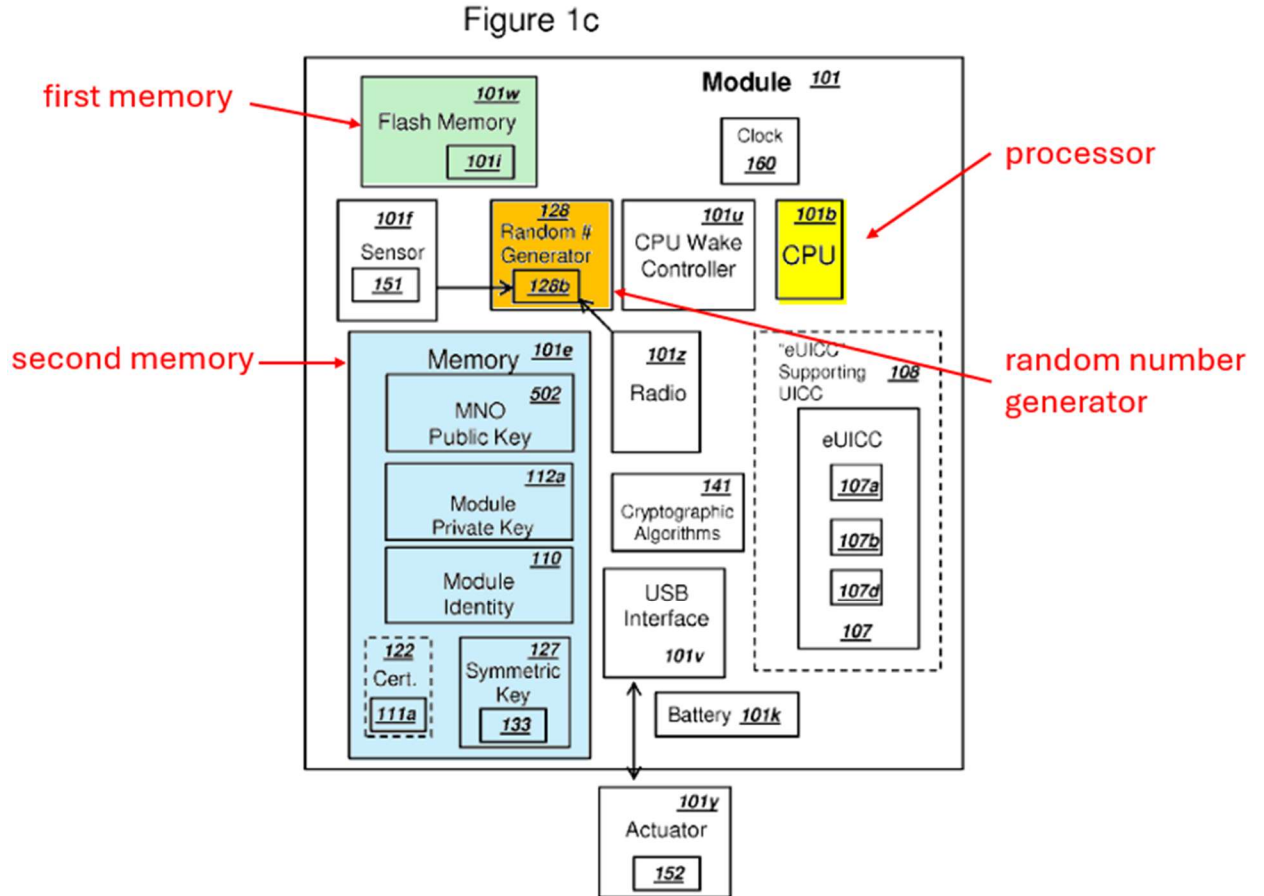
- b. 1[a]: a first memory configured to store an embedded universal integrated circuit card (eUICC) identity;

232. Nix175 teaches “A system for supporting authentication, the system comprising: a nonvolatile *memory* for recording an *embedded Universal Integrated Circuit Card (eUICC) identity*. Ex-1016, Cl. 10.

- c. 1[b]: a random number generator operably connected to a processor connected to a second memory configured to generate a random number for an eUICC private key corresponding to an eUICC public key;

233. Nix175 teaches “As illustrated in FIG. 1c, module 101 may also contain

a random number generator 128 [orange].” Ex-1016, 22:32-33; Fig. 1c. Module 101 also includes flash memory 101w (green) (*id.* 19:24), RAM memory 101e (blue) (*id.* 19:39-41), and a CPU 101b. *Id.* 19:31-33.



234. The random number generator may be used to generate the eUICC private/public key pair: “Several possibilities exist for the source of an eUICC private key 215 and eUICC public key 214 ... The module 101 could derive the PKI key pair using a random number generator 128 and a set of cryptographic algorithms 141, where the random number generator 128 uses input from a sensor 101f and/or

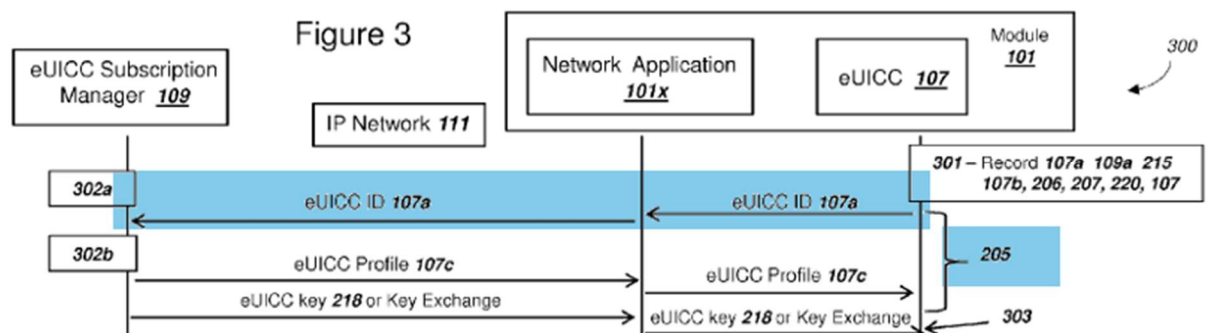
a clock 160 in order to obtain a random number with a high degree of information entropy.” Ex-1016, 42:11-30.

d. 1[c]: a radio including one or more transmit antennas and one or more receiving antennas configured to:

235. Nix175 teaches, “module 101 can comprise a wireless module and include a radio 101z. ... Although not illustrated, radio 101z could include antennas for reception and transmission of RF signals, and even multiple antennas could be used.” 20:42-49.

e. 1[c][1]: a. transmit, to a subscription manager, the eUICC identity and the eUICC public key; and

236. As shown in Figure 3, excerpted below, Nix175’s eUICC transmits the eUICC identity 107a to the subscription manager 109. Ex-1016, 48:65-66.



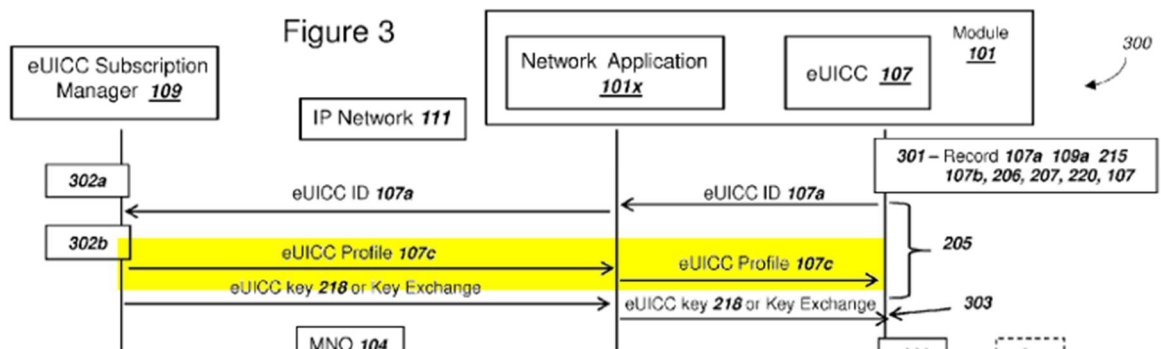
237. Ex-1016, Fig. 3. And “module 101 could send ... send *the eUICC public key 214* along with the eUICC identity 107a (possibly with or in the form of a module identity 110) *to an eUICC subscription manager 109* before the module 101 receives the encrypted eUICC profile 170c.” *Id.* 42:37-42.

- f. 1[c][2]: b. receive, from the subscription manager, i) an eUICC profile comprising network parameters, a key K, and a subscriber identity and ii) a symmetric key; and

238. Nix175-Park-GlobalPlatorm teaches 1[c][2].

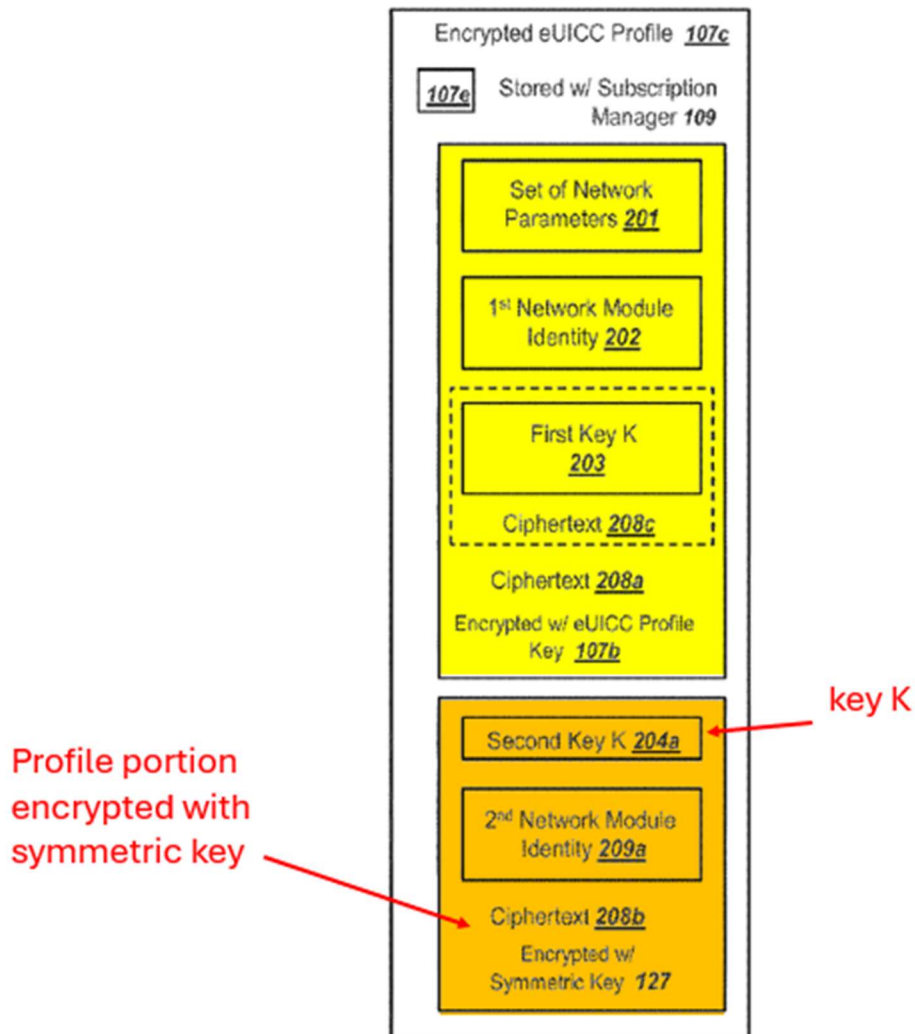
- (1) eUICC profile comprising network parameters, a key K, and a subscriber identity

239. Nix175 shows the module receiving the eUICC profile 107c from the eUICC subscription manager 109 in Figure 3. Ex-1016, Fig. 3.



240. Nix175 further teaches “receiving, by the hardware module, a profile from the eUICC subscription manager, wherein the hardware module uses the profile key to decrypt a first portion of the profile, wherein the first portion includes a first key K1 ... [and] wherein the hardware module uses the symmetric key to decrypt the second portion of the profile, and wherein the second portion includes a **second key K2**.” Ex-1016, Cl. 1. The portion of the profile encrypted with the symmetric key and comprising ciphertext 208b including “**second key K 204a**” (the claimed “key K”) and the **network identity 209a** (claimed “subscriber identity”), is shown in

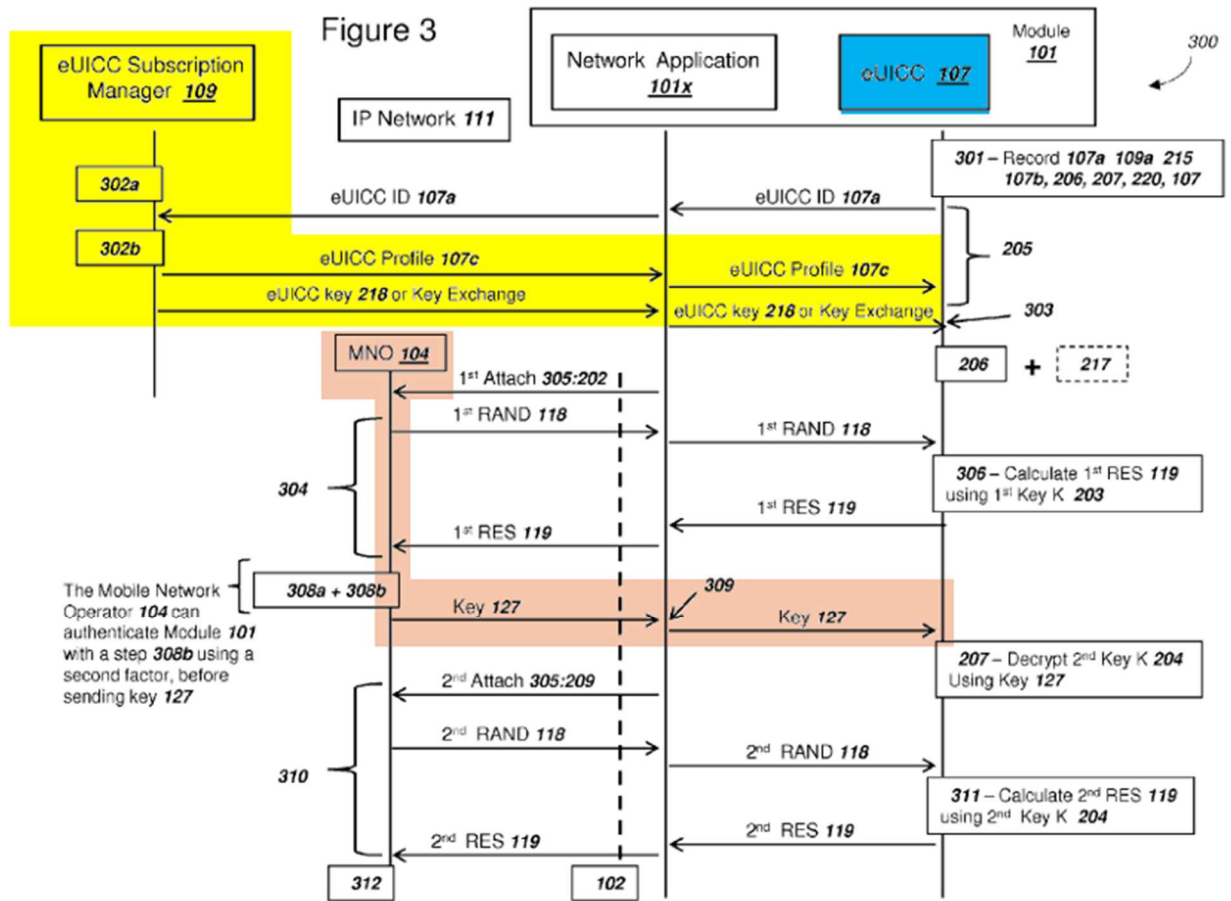
orange in the annotated excerpt of Figure 2a, below. The portion of the profile encrypted with only the profile key and including the *network parameters 201* is shown in yellow.



Ex-1016, Fig. 2a.

(2) symmetric key

241. As discussed above, Nix175 teaches that the *MNO* sends the symmetric key 127 used to encrypt the key K in the profile, as shown below in red.

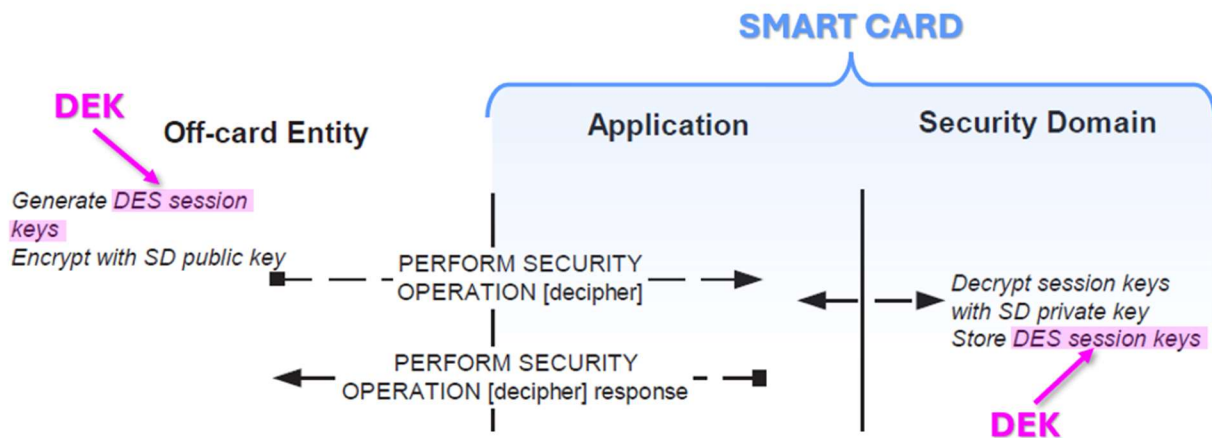


242. But it would have been obvious, in light of Park-GlobalPlatform, that the subscription manager could send the symmetric key 127 to the eUICC, provided that Park’s split architecture of an SM-SR and SM-DP were used to address the security issue discussed above in Section XII.F.1.

243. Specifically Park-GlobalPlatform teaches a “key transport option” that may be used by an off-card entity to send DEK (equivalent to Nix175’s “symmetric key”) to a smart card. Ex-1006, 251-52. The DEK is sent to the card using the “PERFORM SECURITY OPERATION [decipher] command.” Ex-1006, 255-56

(explaining that the PERFORM SECURITY OPERATION [decipher] command includes “control reference templates” that “contain the actual key values,” with “control reference template ‘B8’” used to transmit the DEK). Ex-1006, 247, 275-76.

This is illustrated in Figure F-9, excerpted below:

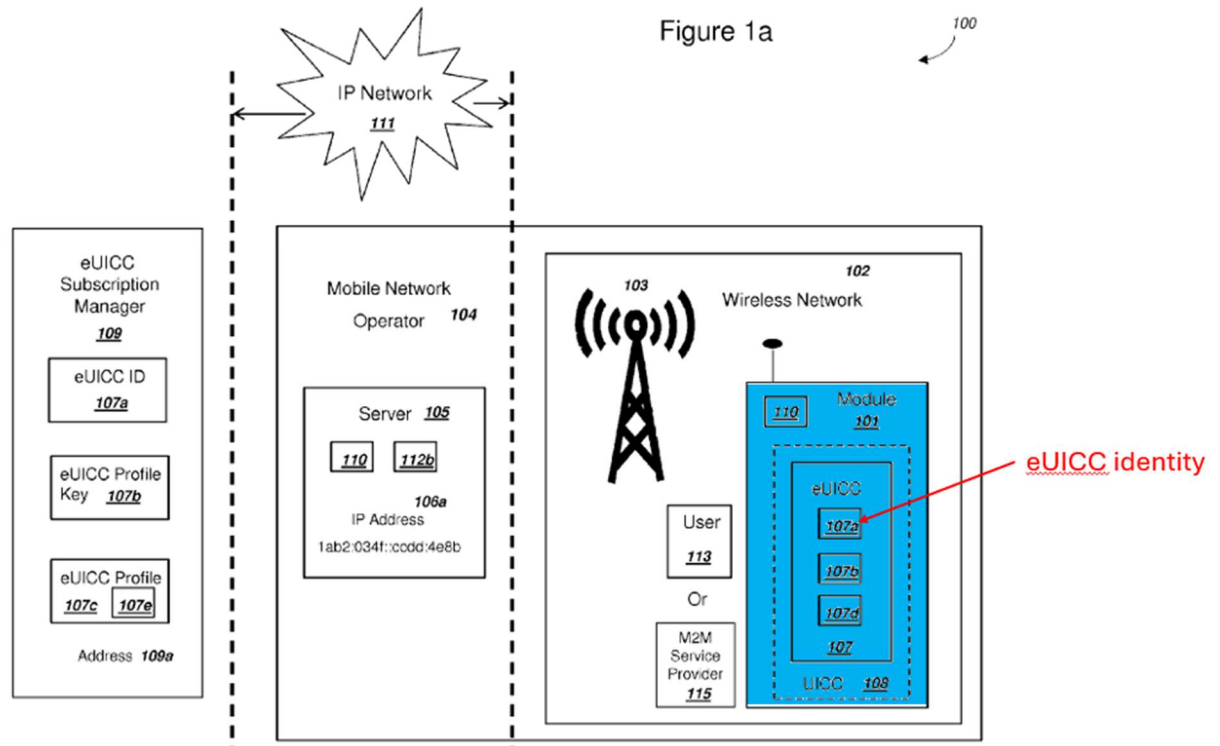


Ex-1006, Fig. F-9 (excerpted)

244. A POSITA would have recognized that in the combined Nix175-Park-GlobalPlatform system, Park’s SM-DP would send the DEK (claimed “symmetric key”) to the eUICC. Further, the SM-DP is part of the network trust zone separate from the SM-SR. Thus, a POSITA would have recognized that in the Nix175-Park-GlobalPlatform system, the subscription manager (component SM-DP) would send the symmetric key to the eUICC without rendering void its security enhancements (as would be the case if the subscription manager of the unmodified Nix175 were to send it).

g. 1[d]: an eUICC associated with the eUICC identity and configured to

245. Nix175 teaches a module 101 having an eUICC 107, wherein the “eUICC 107 can include an eUICC identity 107a.” Ex1016, 11:52-53; Fig. 1a.



h. 1[d][1]: a. derive a profile key using an elliptic curve Diffie-Hellman (ECDH) key exchange with the eUICC private key and a subscription manager public key;

246. Nix175 teaches “An eUICC 107 within a module 101 could use a ECDH key exchange in a step 303 when ECC algorithms are utilized for eUICC public key 214, *eUICC private key* 215, and eUICC subscription manager private key 222 and *eUICC Subscription manager public key* 220.” Ex-1016, 51:65-52:2. Nix175 refers to a 2013 Wikipedia article for a description of the ECDH algorithm

(*id.* 52:4-5), and it was well known that an ECDH algorithm derives a shared key (“profile key”) based on A’s private key and B’s public key. For example:

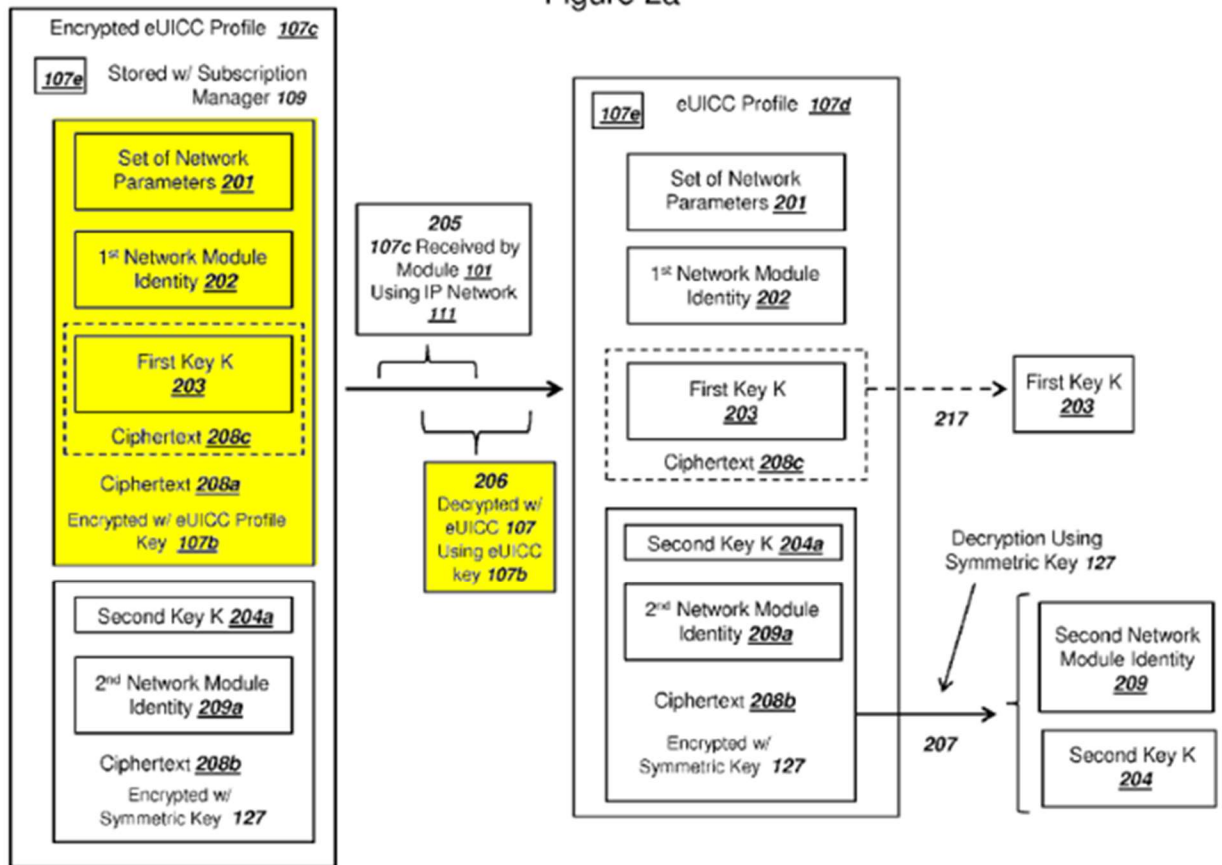
Alice selects a random number n_a and computes $Y_a = n_a B$, she sends Y_a to Bob. Similarly Bob computes $Y_b = n_b B$ and sends Y_b to Alice. Alice and Bob generate the same key $K = n_a Y_b = n_b Y_a = n_a n_b B$.

Ex-1007, 3. The derived key K in this example from Abi-Char corresponds to the claimed “profile key” calculated from A’s (eUICC’s) private key n_a (eUICC private key) and B’s (subscription manager’s) public key Y_b (subscription manager public key).

i. 1[d][2]: b. decrypt a first portion of the eUICC profile using the profile key;

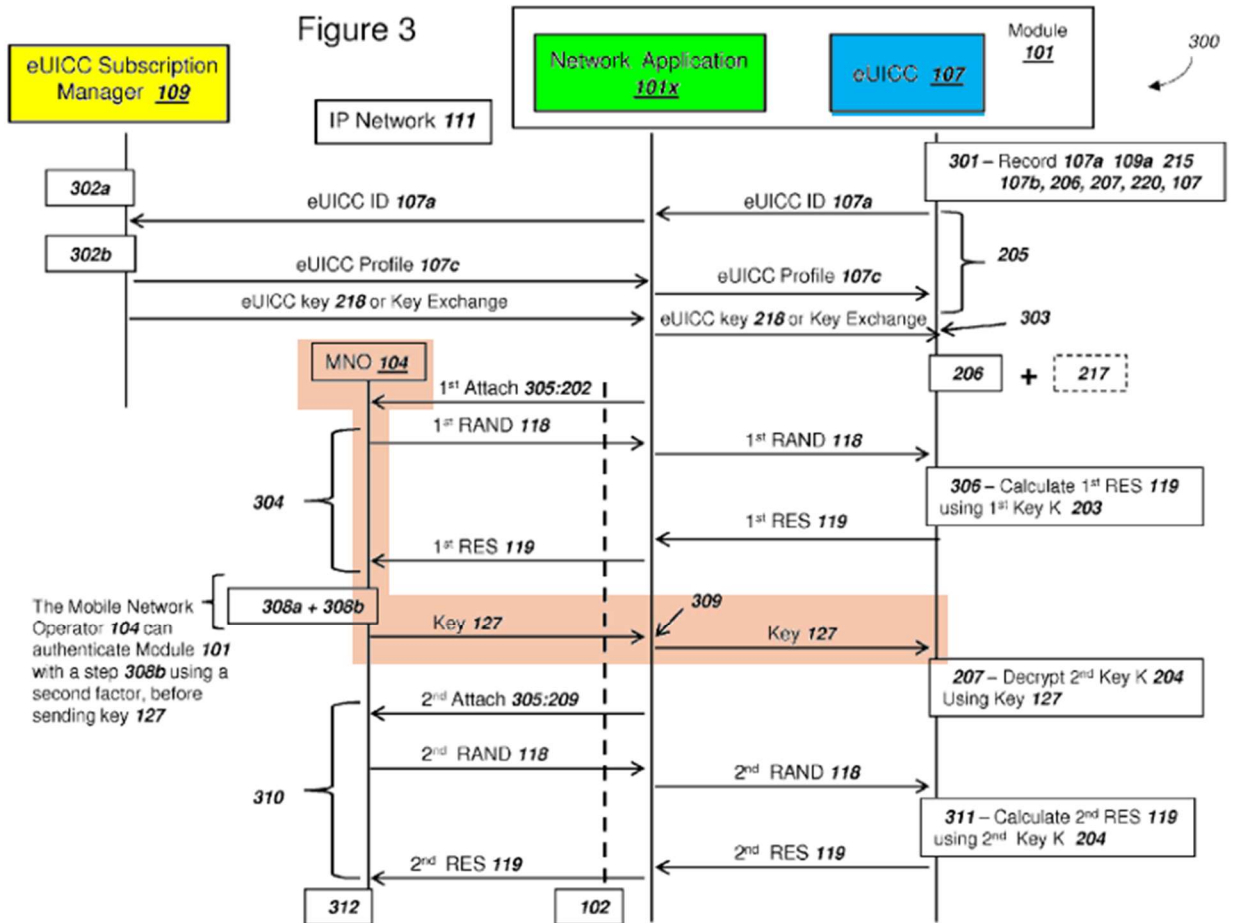
247. Nix175 teaches “receiving, by the hardware module, a profile from the eUICC subscription manager, wherein the hardware module uses the *profile key to decrypt a first portion of the profile.*” Ex-1016, Cl. 1. This is illustrated in the annotated Figure 2a, below, showing the portion of the profile encrypted by the profile key (“eUICC profile key 107b”) in yellow and being decrypted with that key in step 206. Ex-1016, Fig. 2a.

Figure 2a



- j. 1[d][3]: c. receive the symmetric key from a network application operating in the mobile device;

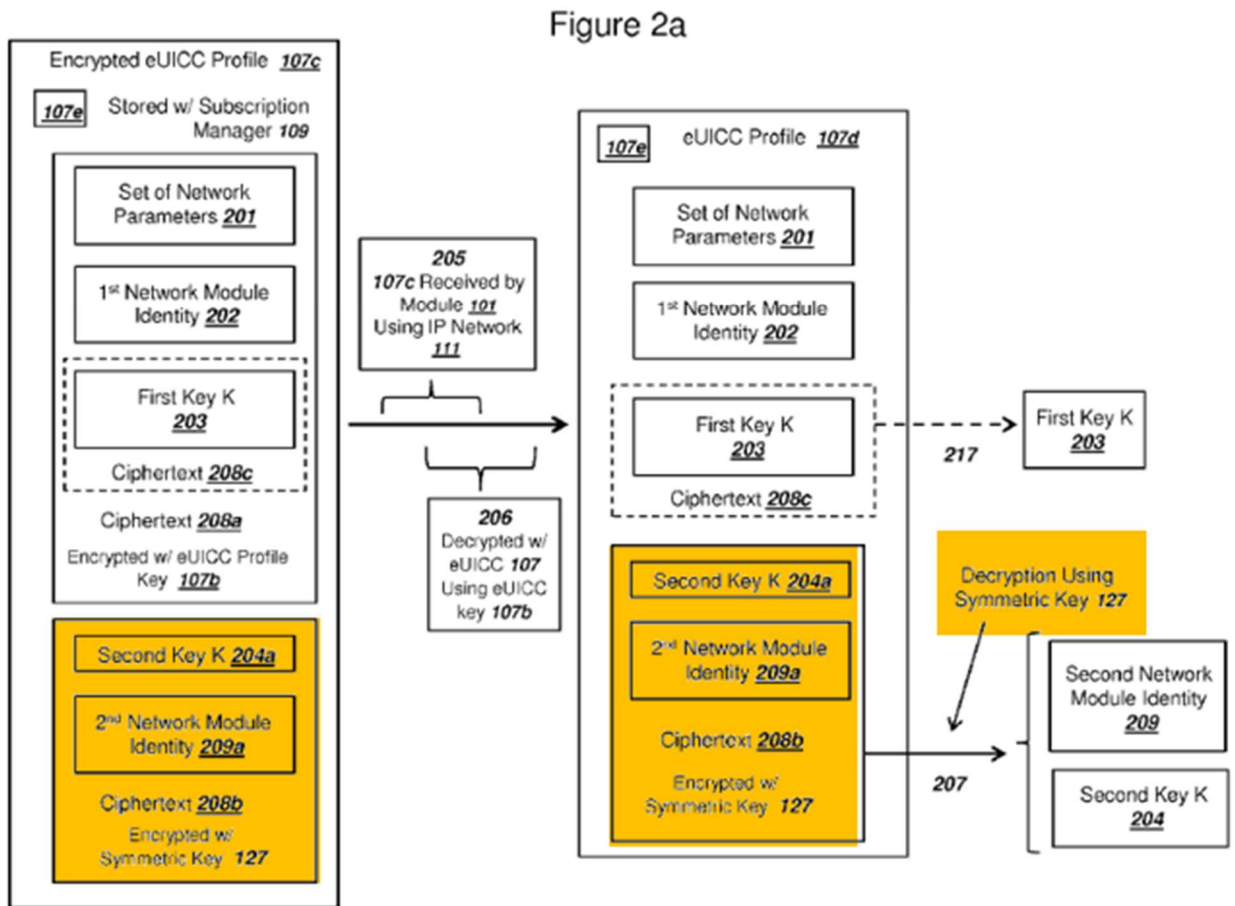
248. Although Nix175's symmetric key is sent by the MNO and not the subscription manager, it is nevertheless received by the module's network application which in turn sends it to the eUICC, as shown in Figure 3, below. MNO 104 (red) sends symmetric key 127 (red) to network application 101x (green), which forwards it on to eUICC 107 (blue). Ex-1016, Fig. 3.



- k. 1[d][4]: d. decrypt a second portion of the eUICC profile using the symmetric key, the second portion comprising the key K and the subscriber identity, wherein the first portion and the second portion are distinct; and

249. Nix175 teaches “wherein the hardware module *uses the symmetric key* to *decrypt the second portion of the profile*, and wherein the second portion includes a second key K2.” Ex-1016, Cl. 1. This is illustrated in Figure 2a below, where orange shading shows the second portion including key 204a and subscriber identity 209a encrypted by symmetric key 127 and then being decrypted by symmetric key

127 at step 207. Ex-1016, Fig. 2a.

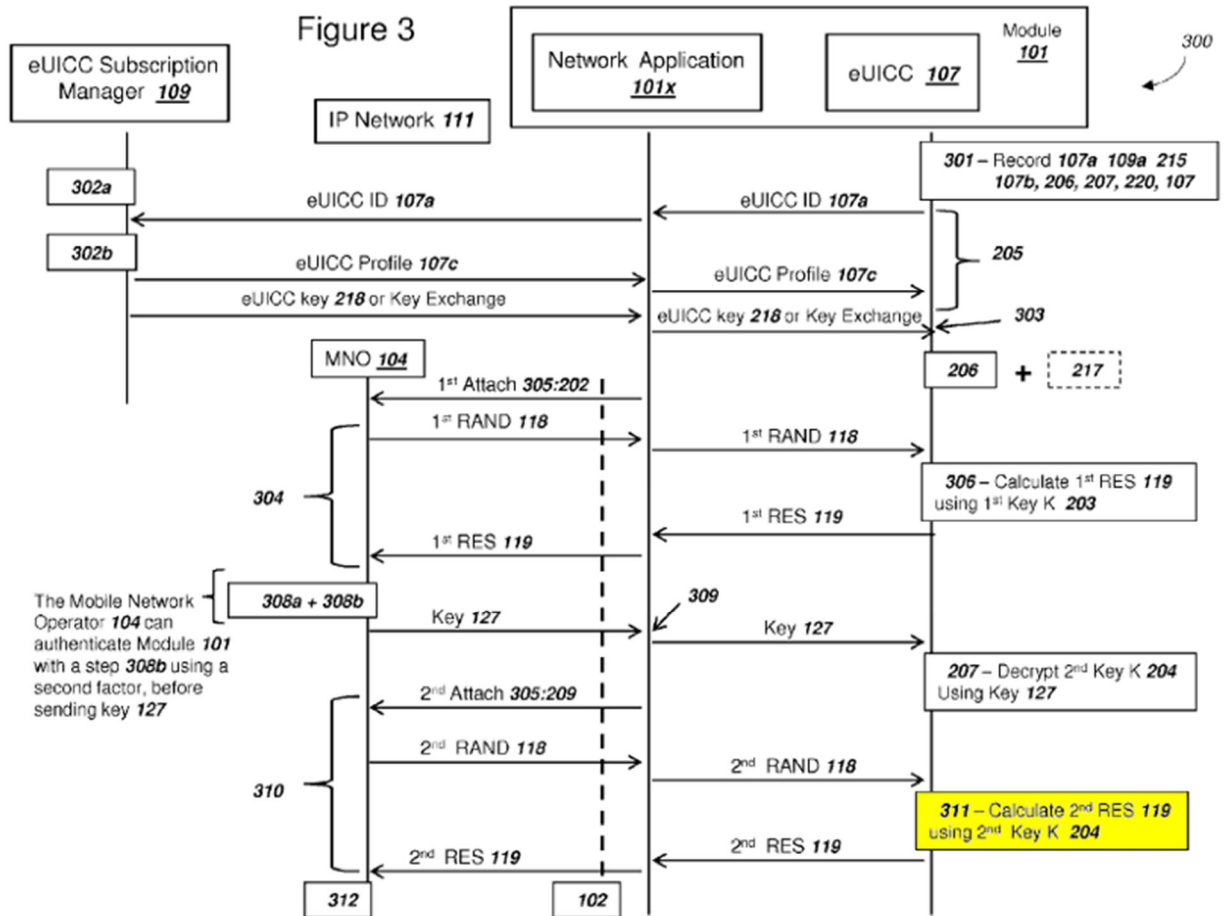


250. Nix175 teaches that ciphertext 208b may be separate from ciphertext 208a, or included within ciphertext 208a. Ex-1016, 33:60-63 (“ciphertext 208b could optionally be included within ciphertext 208a”). As explained in Section YY (motivation to combine), it would have been a simple matter of design choice (computational complexity vs. cryptographic security) to encrypt the sensitive portion only with the symmetric key and the non-sensitive portion only with the profile key, since all of the profile would still be protected under at least one key and

the sensitive portion would remain accessible only to an entity possessing the symmetric key. In either case, ciphertext 208b (second portion of the profile) is distinct from ciphertext 208a (first portion of the profile).

I. 1[d][5]: e. generate a response value for authentication of the mobile device with the wireless network using the key K.

251. Nix175 teaches “At a step 311, the eUICC 107 could calculate the second RES 119 using (i) the second RAND 118 received and (ii) the second key K 204. After receiving the exemplary second RAND 118 message, in order to conduct a second authentication 310, module 101 using an eUICC 107 could take steps to demonstrate to MNO 104 that module 101 has access to the same second key K 204 as recorded by the MNO 104 in an authentication vector 117.” Ex-1016, 60:56-63; Fig. 3.



252. Thus, Nix175-Park-GlobalPlatform renders obvious Claim 1.

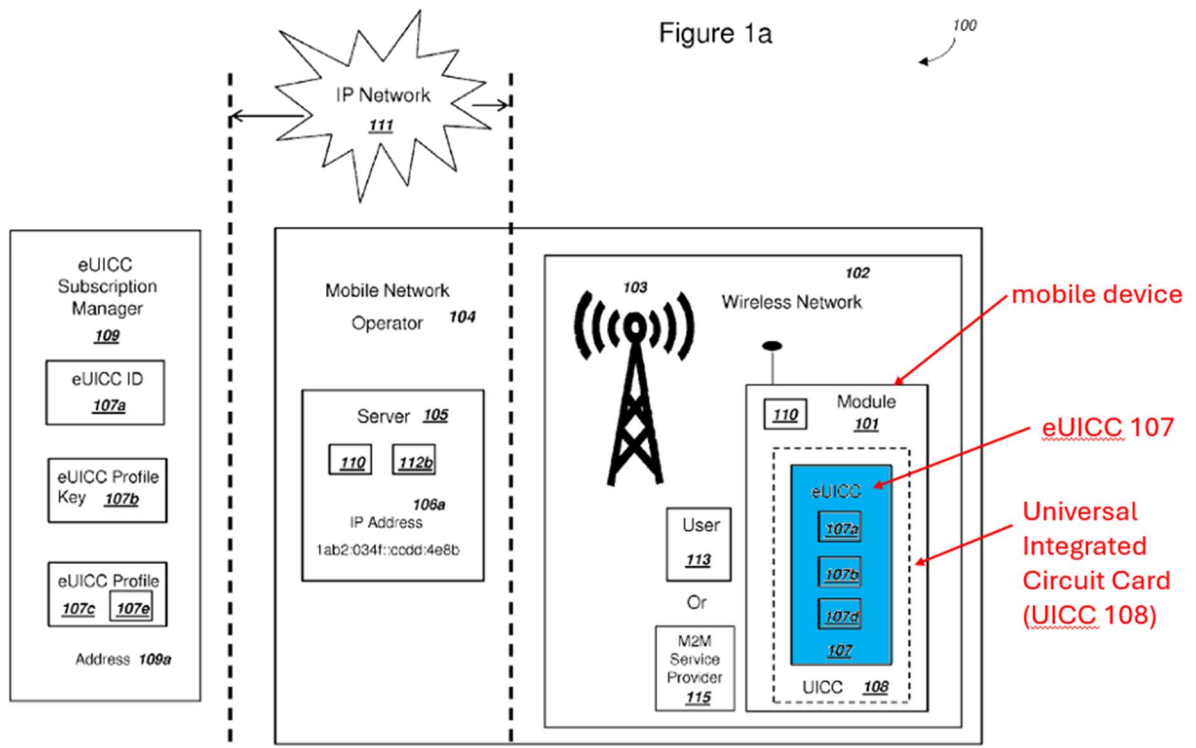
3. Dependent Claims 2-17

- a. **Claim 2: The mobile device of claim 1, wherein the eUICC is further configured to receive a profile identity for the eUICC profile as a plaintext.**

253. Nix175 discloses that the subscription manage transmits the eUICC profile to the eUICC, as discussed above for element 1[c][2]. Nix175 further teaches “Some elements in a profile 107c could remain *plaintext* as well, such as the exemplary *profile identity* 107e.” Ex-1016, 50:47-49.

- b. **Claim 3: The mobile device of claim 1, wherein the eUICC operates within a universal integrated circuit card within the mobile device.**

254. Nix175 teaches an eUICC 107 operating within a UICC 108 within the mobile device 101. Ex-1016, Fig. 1a, below.



- c. **Claim 4: The mobile device of claim 1, wherein the random number generator is configured to generate the random number in response to input from at least one of a clock and a sensor.**

Claim 5: The mobile device of claim 1, wherein the random number generator is configured to generate the random number from a seed value, wherein the seed value comprises data from at least one of a sensor, the radio, a bus, a clock, a physical interface, the memory, and an operating system.

255. Nix175 teaches claims 4 and 5 as follows:

As illustrated in FIG. 1 c, module 101 may also contain *a random number generator 128*. ... A plurality of the data as a source for a random number seed 128b could be appended together into a “module random seed file’ with a combined series or list of states (i.e. a plurality of *sensor 101 f* measurements, *radio 101z* measurements, *clock 160* times or values, memory 101e or memory 101w states, *operating system 101h states*, actuator 101y states, and/or hardware 101a or 101d states).

Ex1016, 22:32-43.

- d. **Claim 6: The mobile device of claim 1, wherein the network application is configured to communicate with i) the wireless network and ii) the eUICC in the mobile device using a system bus.**

256. Nix175 teaches “The *network application 101x* and the eUICC 107 can send and receive data locally within the module 101 using an operating system 101h and/or a *system bus 101d*. ... The module 101, using a *network application 101x*, can send and receive data with the eUICC subscription manager 109 using the *IP network 111*.” Ex-1016, 46:27-35.

- e. **Claim 7: The mobile device of claim 1, wherein the eUICC comprises computer executable instructions for a processor in the mobile device, and wherein the computer executable instructions are stored within the memory.**

257. Nix175 teaches “Many of the logical steps for operation of module 101 or eUICC 107 can be performed in software and hardware,” (Ex-1016, 14:65-66), wherein the eUICC includes “RAM memory, ROM memory, [and] EEPROM memory.” Id. 11:62-12:1. A POSITA would understand one or more of these memories to store the “computer executable instructions,” i.e., software, for performing the operations of eUICC 107.

- f. **Claim 8: The mobile device of claim 1, wherein the eUICC comprises a package soldered onto a circuit board of the mobile device.**

258. Nix175 teaches “An eUICC 107 as illustrated in FIG. 1a and FIG. 1b can be implemented within module 101 in several different ways, including... (iv) a specialized circuit within a surface mount package that is soldered directly onto a circuitboard of the module 101.” Ex-1016, 18:20-32.

- g. **Claim 9: The mobile device of claim 1, further comprising a user interface configured to receive user identification information before the mobile device receives the symmetric key.**

259. Nix175 teaches “the module 101 receives the symmetric key 127 after the MNO 104 authenticates or verifies that a user 113 or M2M service provider 115

is associated with module 101, and until the receipt of the symmetric key 127 the cipher text 208b cannot be feasibly decrypted.” Ex-1016, 58:45-49. Further “wherein a user accesses a user interface of the hardware module to authenticate with the mobile network operator before the hardware module receives the symmetric key.” *Id.* Cl. 2.

260. In the combined system, in which the subscription manager is split into Park’s SM-SR (not necessarily trusted by the MNO) and SM-DP (trusted), the SM-DP would be tasked with authenticating the user, before sending either the symmetric key or the encrypted profile to the user.

h. Claim 10: The mobile device of claim 1, wherein the first portion of the eUICC profile includes the network parameters, the network parameters comprising a list of numbers associated with a mobile network operator.

Claim 11: The mobile device of claim 1, wherein the network parameters comprise a mobile country code (MCC) and a mobile network code (MNC) associated with the wireless network.

261. Nix175 teaches “The profile 107d could include values in the set of network parameters 201 that match or conform with values transmitted by the wireless network 102, such as using the same mobile country code (MCC) and mobile network code (MNC) as recorded in the profile 107d.” Ex-1016, 64:1-5.

- i. **Claim 12: The mobile device of claim 1, wherein the radio is further configured to receive, from the subscription manager, the eUICC profile using transport layer security.**

262. Nix teaches “As transferred across a wireless network 102 and/or IP network 111, the encrypted eUICC profile 107c can be segmented into separate datagrams and transferred using a transport layer protocol Such as TCP.” Ex-1016, 29:16-19.

- j. **Claim 13: The mobile device of claim 1, wherein the subscriber identity comprises an International Mobile Subscriber Identity (IMSI).**

263. Nix175 teaches “In an exemplary embodiment, the first network module identity 202 can comprise an international mobile subscriber identity (IMSI).” Ex-1016, 30:11-13.

- k. **Claim 14: The mobile device of claim 1, wherein the radio is further configured to receive, from the subscription manager, a ciphertext for the symmetric key, wherein the mobile device is configured to decrypt the ciphertext using at least the eUICC private key.**

264. Nix175 teaches “In an exemplary embodiment of a step 309 in FIG. 3, the symmetric key 127 can first be encrypted with a key ciphering algorithm 216. An encrypted symmetric key 127 ciphered with the key ciphering algorithm 216 could be *received by module 101* in a step 309. ... The module 101 could receive the encrypted symmetric key 127 and decrypt the symmetric key 127 in a step 309

by using a key deciphering algorithm 217 *with the eUICC private key 215*” Ex-1016, 58:23-35. However, Nix175 does not teach that it is the “subscription manager” that delivers the symmetric key as ciphertext. But for the same reasons discussed for claim element 1[c][2] above, it would have been obvious in light of Park and GlobalPlatform to receive the ciphertext from Park’s SM-DP.

- l. Claim 15: The mobile device of claim 1, wherein the mobile device further comprises at least one of a wireless handset, a cellular phone, a smartphone, a tablet computer, a laptop, a tracking device, and a circuit board with the radio.**

265. Nix175 teaches “module 101 can be a wireless handset, a cellular phone, a smartphone, a tablet computer, a laptop, a computer with a radio, a tracking device, or a circuit board with a radio that accesses wireless network 102.” Ex-1016, 10:25-29.

- m. Claim 16: The mobile device of claim 1, wherein the second memory is the same memory as the first memory.**

266. Nix175 discloses that that random key data may be stored in either nonvolatile memory (the first memory) or in volatile memory (the second memory), so whether the first and second memory are the same memory is a simple matter of design choice. For example, “The module 101 or eUICC 107 can record the plaintext second key K 204 in a *nonvolatile memory* such as, but not limited to, flash memory *101w*” (*id.* 59:1-3) but “[i]n another embodiment, (A) the plaintext second

key K204 is recorded only in a *volatile memory* within CPU 101b” *Id.* 59:29-30.

- n. **Claim 17: The mobile device of claim 1, wherein the one or more receiving antennas are the one or more transmit antennas.**

267. Nix175 teaches that the antennas can be used for both receive and transmit: “Although not illustrated, radio 101z could include antennas for reception and transmission of RF signals, and even multiple antennas could be used.” Ex-1016, 20:47-49.

XIII. CONCLUSION

268. For the reasons above, it is my opinion that Claims 1-17 of the ’893 Patent are obvious based on each of the grounds specified in this Declaration.

269. I declare that all statements made in this Declaration of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Date: November 26, 2025



Dr. Paul S. Min