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. IPR2025-00118
U.S. Patent No. 12,207,094

DECLARATION OF PAUL S. MIN, PH.D. IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 12,207,094

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 13
a. Encryption Methods 13
b. Key Establishment 16
c. Cryptographic Standards/Requirements 22
2. Authentication 25
3. Integrity and Replay Protection 27
B. GlobalPlatform 30
1. Secure Elements and Security Domains 30
2. Remote Management and Secure Channels 31
C. SIMs and eSIMs/eUICCs 33
VIII. THE CHALLENGED PATENT 39
IX. PRIORITY DATE 45
X. CLAIM CONSTRUCTION 46
XI. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART 47
A. Park (Ex-1005) 47
B. GlobalPlatform (Ex-1006) 48

TABLE OF CONTENTS
(continued)

Page

C. AbiChar (Ex-1007) 51

D. X9.63-Overview (Ex-1008) 51

E. Weiss (Ex-1009) 53

F. Nix175 (Ex-1016) 53

G. SGP.22 (Ex-1010) 54

H. SCP11 (Ex-1011) 54

XII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS 55

A. Ground 1: Claims 1-17 and 19-22 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 61

2. Independent Claim 1 64

a. Element 1[pre]: A method comprising: 64

b. Element 1[a][1]: (a) generating a first message comprising 64

c. Element 1[a][2][i]: (1) an identity of an embedded universal integrated circuit card (eUICC); 65

d. Element 1[a][2][ii]: (2) a nonce; and 66

Petitioner Samsung Ex-1002, 0003

TABLE OF CONTENTS
(continued)

Page

e. Element 1[a][2][iii]: (3) a first digital signature, generated using a first eUICC private key, wherein the first eUICC private key corresponds to a first eUICC public key; 68

f. Element 1[b]: (b) sending the first message via a network application to a subscription manager system; 69

g. Element 1[c]: (c) deriving a second eUICC private key and a corresponding second eUICC public key using a first random number generator and a first set of cryptographic algorithms; 70

h. Element 1[d]: (d) storing a subscription manager public key which corresponds to a subscription manager private key; 71

i. Element 1[e][1]: (e) deriving a profile key using a key exchange algorithm based on at least: 72

j. Element 1[e][2]: wherein the profile key can also be derived at the subscription manager system based at least on: the second eUICC public key, and the subscription manager private key; 74

k. Element 1[f]: receiving from the subscription manager system, an encrypted profile comprising a ciphertext including a key K encrypted with a symmetric key;.....74

l. Element 1[g]: (g) receiving the symmetric key;.....76

m. Element 1[h]: (h) decrypting, by the embedded universal integrated circuit card, at least a portion of the encrypted profile using the profile key; 77

n. Element 1[i]: (i) decrypting, by the embedded universal integrated circuit card, at least a portion of the ciphertext using the symmetric key; and.....78

Petitioner Samsung Ex-1002, 0004

TABLE OF CONTENTS
(continued)

Page

o. Element 1[j]: (j) storing at least the key K in the embedded universal integrated circuit card for use in future communications. 79

3. Dependent Claims 2-17, 19-22 79

a. Claim 2: The method of claim 1, wherein the subscription manager public key is downloaded via the network application. 79

b. Claim 3: The method of claim 1, wherein the step of storing the subscription manager public key occurs at the time of manufacture of the embedded universal integrated circuit card. 80

c. Claim 4: The method of claim 1, wherein the key exchange algorithm is a Diffie-Hellman key exchange. 82

d. Claim 6: The method of claim 1, wherein the encrypted profile is encrypted in a manner so that intermediate nodes on an IP network would not be able to read data within the profile in an unencrypted form.....82

e. Claim 7: The method of claim 1, wherein subscription manager system stores the ciphertext and the symmetric key before step (a).....83

f. Claim 8: The method of claim 1, wherein after the step (b), the embedded universal integrated circuit card is authenticated by the subscription manager system using at least the identity of the embedded universal integrated circuit card included in the first message. 86

g. Claim 10: The method of claim 1, wherein the decrypted ciphertext is stored in nonvolatile memory of the embedded universal integrated circuit card. 88

Petitioner Samsung Ex-1002, 0005

TABLE OF CONTENTS
(continued)

Page

h. Claim 11: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module and wherein the module is a machine-to-machine module.....88

i. Claim 12: The method of claim 11, wherein the module comprises a module of a type selected from the group consisting of: vending machine, automobile, alarm system, tracking device, monitoring device, security system and remote control device.....89

j. Claim 13: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module and wherein the module is a wireless module.....89

k. Claim 15: The method of claim 13, wherein the module comprises a module of a type selected from the group consisting of the following: wireless handset, cellular phone, smartphone, laptop computer, tracking device, computer with a radio, circuit board with a radio, hand-held device, netbook, portable computer, multiprocessor system, microprocessor based consumer electronic, programmable consumer electronic, network personal computer, minicomputer, mainframe computer and smartcard.....90

l. Claim 16: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module, wherein the module is a wired module.....91

m. Claim 19: The method of claim 1, wherein the deriving step (c) is performed prior to the generating step (a).....92

Petitioner Samsung Ex-1002, 0006

TABLE OF CONTENTS
(continued)

Page

n. Claim 21: The method of claim 1, wherein the embedded integrated circuit card is operably connected to a module and a user accesses a user interface of the module to enter authentication information before the module receives the symmetric key.....93

o. Claim 22: The method of claim 1, wherein the subscription manager receives the ciphertext from a mobile network operator before the network application sends the first message.....94

B. Ground 2: Claims 1-17 and 19-22 are obvious over Park in view of GlobalPlatform and X9.63-Overview.....94

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.....94

2. Independent Claim 1 96

a. Elements 1[pre]-1[b] and 1[f]-1[j].....96

b. Element 1[c]: (c) deriving a second eUICC private key and a corresponding second eUICC public key using a first random number generator and a first set of cryptographic algorithms;.....96

c. Element 1[d]: (d) storing a subscription manager public key which corresponds to a subscription manager private key; 98

d. Element 1[e][1]: (e) deriving a profile key using a key exchange algorithm based on at least: 99

e. Element 1[e][2]: wherein the profile key can also be derived at the subscription manager system based at least on: the second eUICC public key, and the subscription manager private key;.....100

3. Dependent Claims 2-17, 19-22 100

Petitioner Samsung Ex-1002, 0007

TABLE OF CONTENTS
(continued)

Page

a. Claims 2-3, 6-17, 19-22 100

b. Claim 4: The method of claim 1, wherein the key exchange algorithm is a Diffie-Hellman key exchange. 100

C. Grounds 3-4: Claim 18 is obvious over Park in view of GlobalPlatform, AbiChar, and Weiss, and Park in view of GlobalPlatform, X9.63-Overview, and Weiss 101

1. A POSITA would have been motivated to add Weiss's teachings to the Park-GlobalPlatform-AbiChar and Park-GlobalPlatform-X9.63-Overview systems and would have had a reasonable expectation of success. 101

2. Claim 18: The method of claim 16, wherein the module comprises a module of a type selected from the group consisting of: computer, security camera, security monitoring device and networked controller. 102

D. Ground 5: Claims 1-22 are obvious over Nix175 in view of Park and GlobalPlatform 102

1. A POSITA would have been motivated to combine Nix175, Park, and GlobalPlatform and would have had a reasonable expectation of success. 102

2. Independent Claim 1 108

a. Element 1[pre]: A method comprising: 108

b. Element 1[a][1]: (a) generating a first message comprising (1) an identity of an embedded universal integrated circuit card (eUICC); (2) a nonce; and (3) a first digital signature, generated using a first eUICC private key, wherein the first eUICC private key corresponds to a first eUICC public key; 108

c. Element 1[b]: (b) sending the first message via a network application to a subscription manager system; 109

Petitioner Samsung Ex-1002, 0008

TABLE OF CONTENTS
(continued)

Page

d. Element 1[c]: (c) deriving a second eUICC private key and a corresponding second eUICC public key using a first random number generator and a first set of cryptographic algorithms; 110

e. Element 1[d]: (d) storing a subscription manager public key which corresponds to a subscription manager private key; 111

f. Element 1[e][1]: (e) deriving a profile key using a key exchange algorithm based on at least: 112

(i) the second eUICC private key, and..... 112

(ii) the stored subscription manager public key,..... 112

g. Element 1[e][2]: wherein the profile key can also be derived at the subscription manager system based at least on: the second eUICC public key, and the subscription manager private key; 113

h. Element 1[f]: (f) receiving from the subscription manager system, an encrypted profile comprising a ciphertext including a key K encrypted with a symmetric key;..... 113

i. Element 1[g]: (g) receiving the symmetric key;..... 114

j. Element 1[h]: (h) decrypting, by the embedded universal integrated circuit card, at least a portion of the encrypted profile using the profile key; 117

k. Element 1[i]: (i) decrypting, by the embedded universal integrated circuit card, at least a portion of the ciphertext using the symmetric key; and..... 118

l. Element 1[j]: (j) storing at least the key K in the embedded universal integrated circuit card for use in future communications. 119

3. Dependent Claims 2-22..... 119

Petitioner Samsung Ex-1002, 0009

TABLE OF CONTENTS
(continued)

Page

a. Claim 2: The method of claim 1, wherein the subscription manager public key is downloaded via the network application. 119

b. Claim 3: The method of claim 1, wherein the step of storing the subscription manager public key occurs at the time of manufacture of the embedded universal integrated circuit card. 120

c. Claim 4: The method of claim 1, wherein the key exchange algorithm is a Diffie-Hellman key exchange. 121

Claim 5: The method of claim 1, wherein the key exchange algorithm is an Elliptic Curve Diffie-Hellman key exchange.....121

d. Claim 6: The method of claim 1, wherein the encrypted profile is encrypted in a manner so that intermediate nodes on an IP network would not be able to read data within the profile in an unencrypted form.....121

e. Claim 7: The method of claim 1, wherein subscription manager system stores the ciphertext and the symmetric key before step (a).....122

f. Claim 8: The method of claim 1, wherein after the step (b), the embedded universal integrated circuit card is authenticated by the subscription manager system using at least the identity of the embedded universal integrated circuit card included in the first message. 123

Claim 9: The method of claim 1, wherein after the step (b), the embedded universal integrated circuit card is authenticated by the subscription manager system using at least the digital signature included in the first message. 123

Petitioner Samsung Ex-1002, 0010

TABLE OF CONTENTS
(continued)

Page

g. Claim 10: The method of claim 1, wherein the decrypted ciphertext is stored in nonvolatile memory of the embedded universal integrated circuit card. 123

h. Claim 11: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module and wherein the module is a machine-to-machine module.....124

i. Claim 12: The method of claim 11, wherein the module comprises a module of a type selected from the group consisting of: vending machine, automobile, alarm system, tracking device, monitoring device, security system and remote control device.....124

j. Claim 13: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module and wherein the module is a wireless module.....125

Claim 14: The method of claim 13, wherein the module is configured to communicate using one or more wireless technologies of types selected from the group consisting of: WiFi, WiMax, General Packet Radio Services, Enhanced Data rates for GSM Evolution and 3rd Generation Partnership Project technology.....125

Petitioner Samsung Ex-1002, 0011

TABLE OF CONTENTS
(continued)

Page

k. Claim 15: The method of claim 13, wherein the module comprises a module of a type selected from the group consisting of the following: wireless handset, cellular phone, smartphone, laptop computer, tracking device, computer with a radio, circuit board with a radio, hand-held device, netbook, portable computer, multiprocessor system, microprocessor based consumer electronic, programmable consumer electronic, network personal computer, minicomputer, mainframe computer and smartcard.....126

l. Claim 16: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module, wherein the module is a wired module.....126

Claim 17: The method of claim 16, wherein the module is configured to communicate using one or more wired connections of types selected from the group consisting of: Ethernet, fiber optic connection and Universal Serial Bus connection. 126

m. Claim 18: The method of claim 16, wherein the module comprises a module of a type selected from the group consisting of: computer, security camera, security monitoring device and networked controller.....127

n. Claim 19: The method of claim 1, wherein the deriving step (c) is performed prior to the generating step (a).....127

Claim 20: The method of claim 1, wherein the generating step (a) is performed before the deriving step (c). 127

Petitioner Samsung Ex-1002, 0012

TABLE OF CONTENTS
(continued)

Page

o. Claim 21: The method of claim 1, wherein the embedded integrated circuit card is operably connected to a module and a user accesses a user interface of the module to enter authentication information before the module receives the symmetric key.....127

p. Claim 22: The method of claim 1, wherein the subscription manager receives the ciphertext from a mobile network operator before the network application sends the first message.....128

XIII. CONCLUSION.....128

Petitioner Samsung Ex-1002, 0013

LIST OF EXHIBITS[1]

Ex. No.	Description
Ex-1001	U.S. Patent No. 12,207,094 ("the '094 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. 12,207,094
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	U.S. Patent Pub. No. 2015/0350881 to Weiss ("Weiss")
Ex-1010	GSMA SGP.22 – RSP Technical Specification, Version 2.0 (Oct. 2016) ("SGP.22")

[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-1011	GlobalPlatform Card Secure Channel Protocol '11' Card Specification v2.2 – Amendment F, Version 1.0 (May 2015) ("SCP11")
Ex-1012	INTENTIONALLY LEFT BLANK
Ex-1013	INTENTIONALLY LEFT BLANK
Ex-1014	INTENTIONALLY LEFT BLANK
Ex-1015	Declaration of Simon Blake-Wilson, author of ANSI X9.63-Overview
Ex-1016	INTENTIONALLY LEFT BLANK
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	Diffie and Hellman, "New Directions in Cryptography" IEEE Transactions on Information Theory, Vol. 22, No. 6 (November 1976)
Ex-1023	NIST Special Publication 800-57, Recommendation for Key Management – Part 1: General (Rev. 3) (July 2012)
Ex-1024	NIST Special Publication 800-38F, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping (December 2012)

Ex. No.	Description
Ex-1025	NIST Special Publication 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 (April 2009)
Ex-1027	Reprogrammable SIMs: Technology, Evolution and Implications, Final Report, Ofcom (September 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 '094 Patent, filed on September 30, 2025, as Exhibit F to Network-1 Technologies, Inc.'s P.R. 3-1 Disclosure of Asserted Claims and Infringement Contentions
Ex-1030	U.S. Patent No. 11,916,893

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. 12,207,094 ("'094 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 1-22 of the '094 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, S3GPP C-FDMA, and 3GPP TDD. In unitizing 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 '094 Patent, the references cited by the '094 Patent, the prosecution history of the '094 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 **Error! Reference source not found.**.

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 ’094 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 ’094 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 ’094 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 cyber-security 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 1-22 of the ’094 Patent would have been obvious under 35 U.S.C. § 103 on the following grounds:

- **Ground 1:** Claims 1-17, and 19-22 are obvious over Park (Ex-1005) in view of GlobalPlatform (Ex-1006) and AbiChar (Ex-1007).
- **Ground 2:** Claims 1-17, and 19-22 are obvious over Park in view of GlobalPlatform and X9.63-Overview (Ex-1008).
- **Ground 3:** Claim 18 is obvious over Park in view of GlobalPlatform, AbiChar, and Weiss (Ex-1009).
- **Ground 4:** Claim 18 is obvious over Park in view of GlobalPlatform, X9.63-Overview, and Weiss.
- **Ground 5:** Claims 1-22 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.

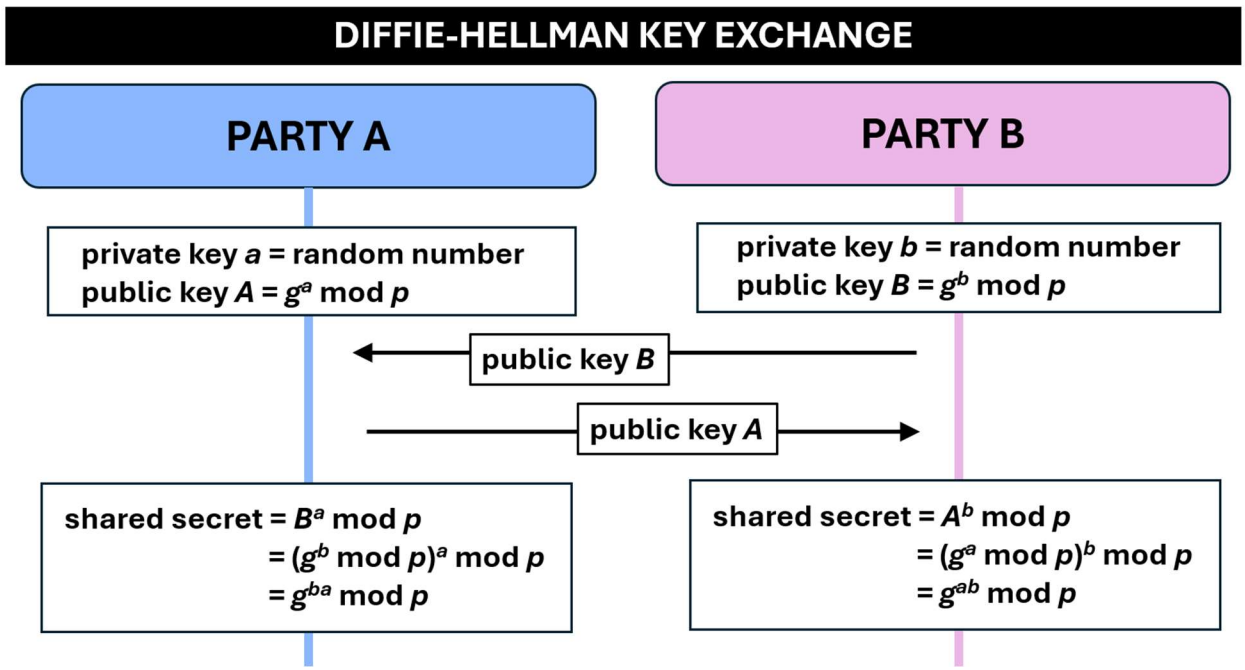
(2) Each party chooses a private random number (a and b, respectively), and then computes a corresponding public key ($A = g^a \bmod p$ and $B = g^b \bmod p$, respectively). Here, the modulus ("mod") operation means that after computing g^a or g^b, the result is divided by p, and only the remainder is 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, $S = g^{ab} \bmod p$.

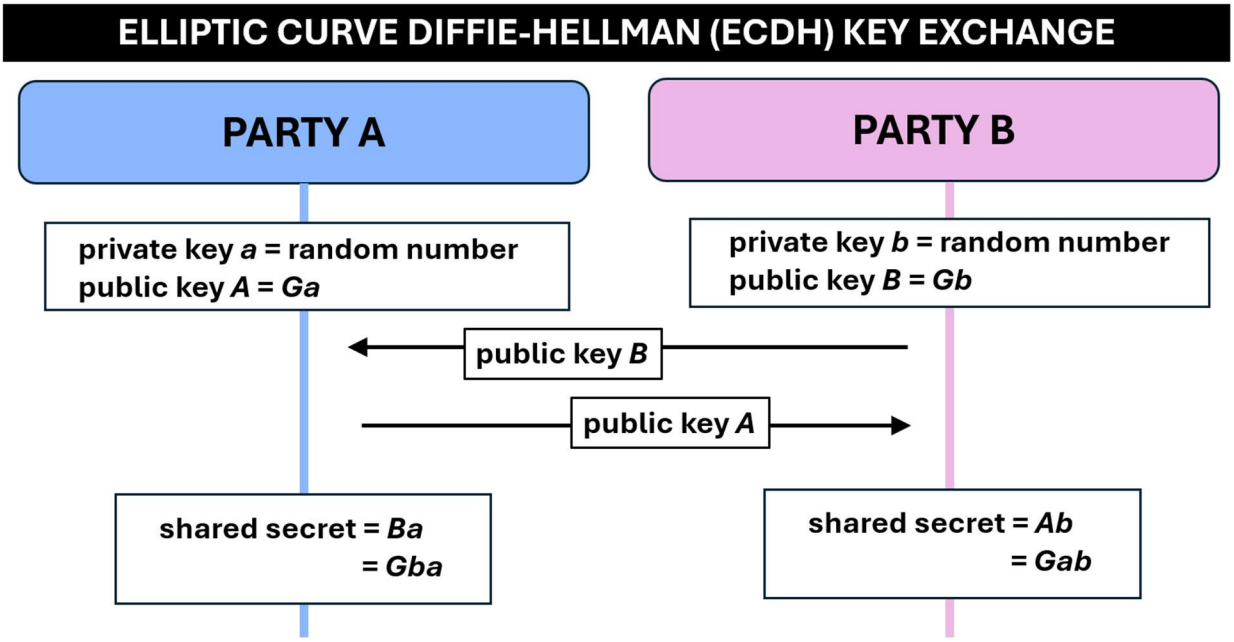
See, e.g., Ex-1021, Boyd & Mathuria, at 141-42. These steps are illustrated in the diagram below:



(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:

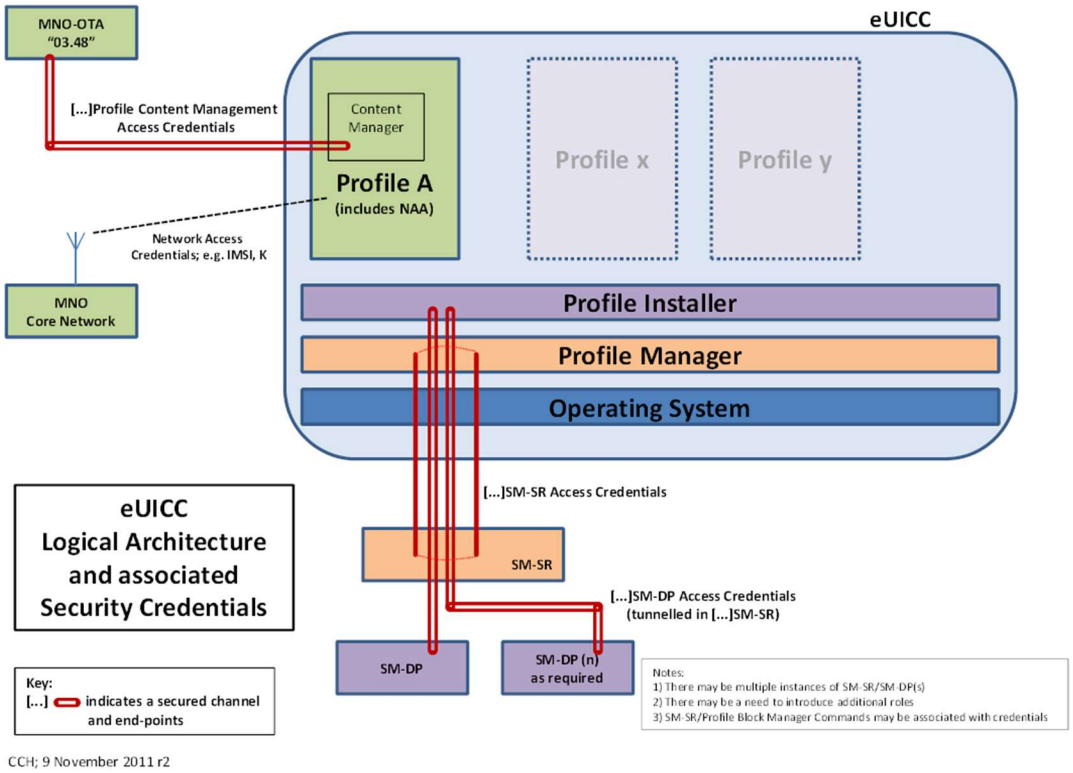
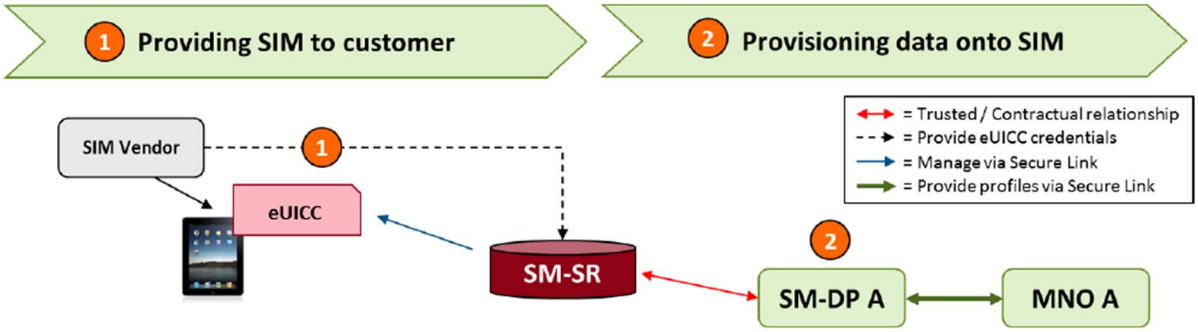


Figure C.1

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:

Figure 1: eUICC Provisioning Ecosystem



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.

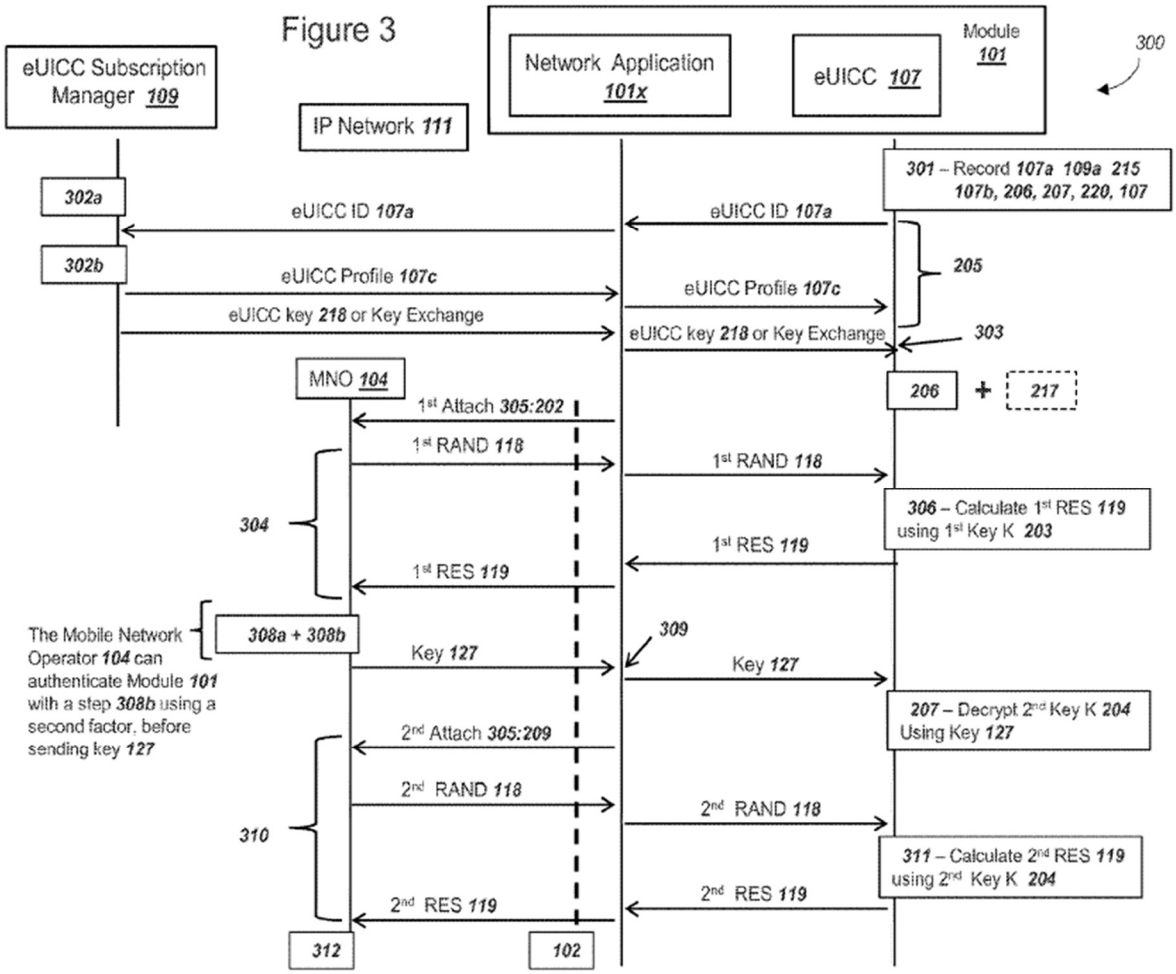
VIII. THE CHALLENGED PATENT

84. The ’094 Patent relates to techniques for provisioning an embedded universal integrated circuit card (eUICC) with a profile containing network access credentials. Ex-1001, 3:62-4:4.

85. As the ’094 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, 2:41-52. 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, 2:52-56. 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, 3:8-3:22. 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, 3:23-27.

86. The ’094 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, 3:66-4:4, 35:45-56. To address this issue, the '780 Patent describes a multi-stage provisioning process illustrated in Figure 3, below, splitting control between the SM (for delivering the secure profile) and the MNO (for transmitting the symmetric key used to encrypt secret key K).[2]



[2] Throughout, emphasis and annotations are added unless otherwise specified.

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.*, 5:42-49, 6:47-49. The eUICC receives this profile from the SM, encrypted under a "profile key" that is known to both the SM and the eUICC. *Id.*, 5:41-47. 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.*, 5:52-6:2; Fig. 3, 308a, 308b. After a second-factor authentication, the MNO transmits the symmetric key needed to decrypt the second portion of the profile. *Id.*, 6:37-39. 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.*, 6:50-7:9.

88. Thus, the purpose of the '094 Patent is its two-stage provisioning process, which ensures that the network 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) Elements (a)-(b) recite sending a profile provisioning request with information that can be used to authenticate the eUICC and using a first eUICC private key to create a digital signature; (2) Elements (c)-(e) recite participating in an asymmetric key exchange using a second derived eUICC

private/public key pair to generate a shared "profile key" for use in encrypting/decrypting the profile; and (3) Elements (f)-(j) recite receiving and decrypting the 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 second portion of the profile that includes a network access key K. Ex-1001, Cl. 1.

1. A method comprising:

(a) generating a first message comprising:
(1) an identity of an embedded universal integrated circuit card (eUICC);
(2) a nonce; and
(3) a first digital signature, generated using a first eUICC private key, wherein the first eUICC private key corresponds to a first eUICC public key;
(b) sending the first message via a network application to a subscription manager system;

profile provision request with information needed to authenticate the eUICC

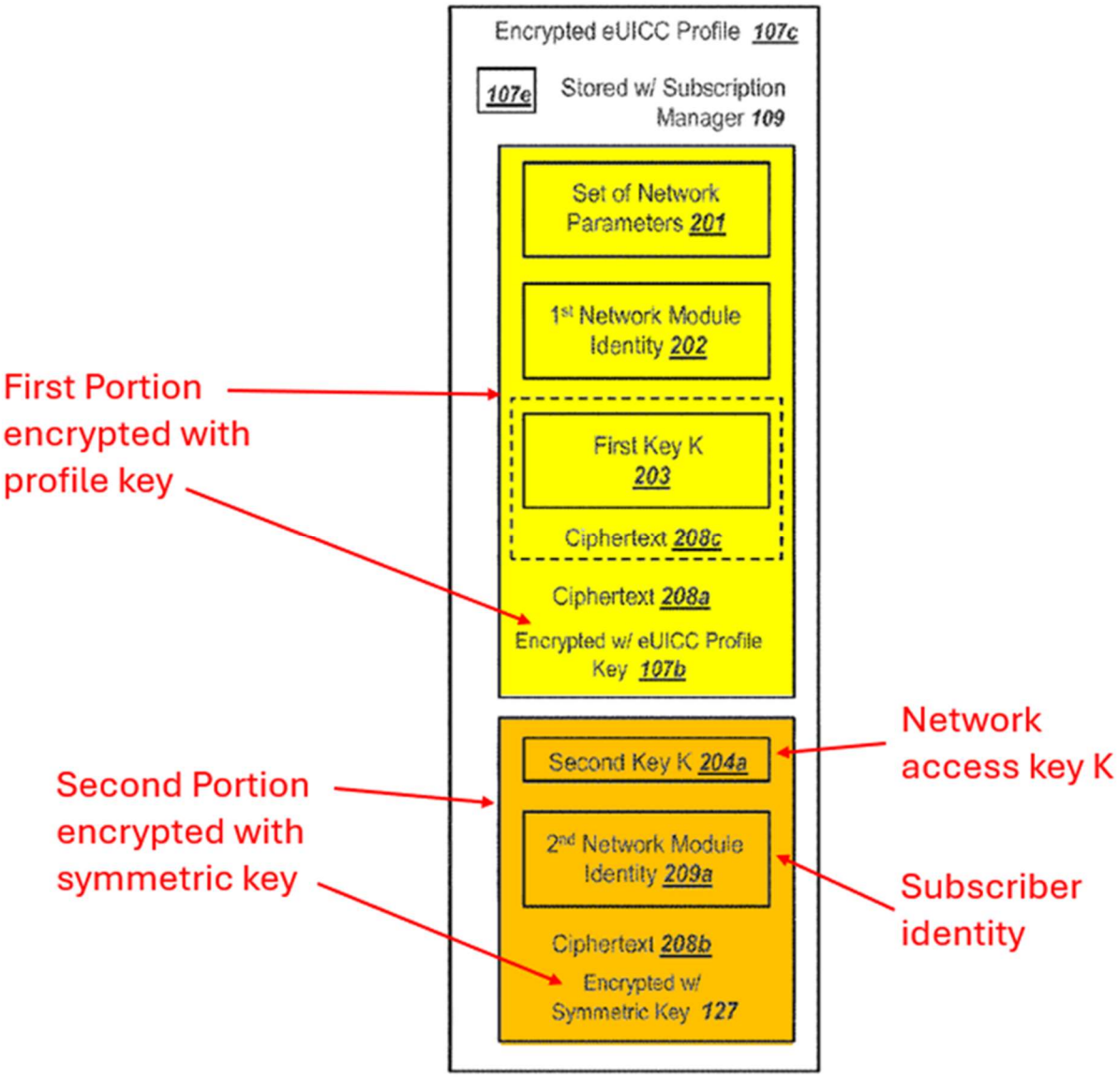
(c) deriving a second eUICC private key and a corresponding second eUICC public key using a first random number generator and a first set of cryptographic algorithms;
(d) storing a subscription manager public key which corresponds to a subscription manager private key;
(e) deriving a profile key using a key exchange algorithm based on at least:
(i) the second eUICC private key, and
(ii) the stored subscription manager public key, wherein the profile key can also be derived at the subscription manager system based at least on: the second eUICC public key, and the subscription manager private key;

perform an asymmetric key exchange with an SM to derive a shared, symmetric "profile key"

(f) receiving from the subscription manager system, an encrypted profile comprising a ciphertext including a key K encrypted with a symmetric key;
(g) receiving the symmetric key;
(h) decrypting, by the embedded universal integrated circuit card, at least a portion of the encrypted profile using the profile key;
(i) decrypting, by the embedded universal integrated circuit card, at least a portion of the ciphertext using the symmetric key; and
(j) storing at least the key K in the embedded universal integrated circuit card for use in future communications.

receive encrypted profile having an encrypted, second portion, that contains secret key K, and is decrypted using a separately received symmetric key

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



90. The ’094 Patent was filed January 18, 2024, and claims priority through a chain of 6 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 double patenting and Section 112 rejections,

which the Applicant overcame by filing terminal disclaimers and amending the claims to provide antecedent basis for the claimed eUICC. Ex-1004.

IX. PRIORITY DATE

91. I understand that the ’094 Patent alleges the earliest priority date of December 6, 2013. However, for two reasons, the claims of the ’094 Patent are broader than the disclosure of the earliest priority application, filed on December 6, 2013.

92. First, while element 1(a)(3) recites a “first eUICC public key” and element 1(c) recites a “second eUICC public key,” the specification discloses only one eUICC private/public key pair: eUICC private key 215 and eUICC public key 214 and does not disclose two separate pairs. *E.g.*, Ex-1001, 43:36-67; Ex-1016, 42:11-42. Second, the common specification of the ’094 Patent and its predecessors make 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, 6:37-39, 35:19-45, 35:56-62, 41:65-42:13, 59:44-55, 59:62-60:6. However, the claims of the ’094 Patent place no limit on the source of the claimed “symmetric key,” and, in the parties’ district court litigation, PO has taken the position that the eUICC receives the symmetric key not from the MNO, but from the

subscription manager. Ex-1029 (Network-1's infringement contentions for the '094 Patent).

93. That the symmetric key may be sent from the subscription manner, rather than the MNO, was first disclosed within the '094 Patent's priority chain in the claims of the '893 Patent. Ex-1030, Cl. 1 (reciting "receive, from the subscription manager, … a symmetric key"). I understand from counsel that a patent is only entitled to the priority date of an earlier-filed application in the patent's priority chain, if that earlier-filed application provides written description support for the full scope of its claims. Accordingly, the '094 Patent is entitled to a priority date no earlier than December 10, 2021, the filing date of the earlier '893 Patent.

94. Ground 5 assumes this later priority date, while the references relied upon in Grounds 1-4 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 '094 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.

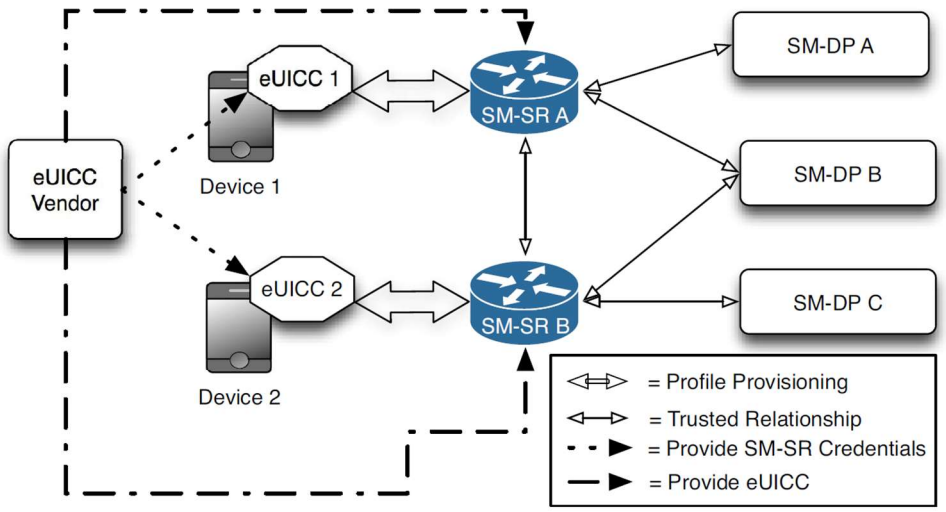


Fig. 1. eUICC Profile Ecosystem

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 web site. Ex-1018. I understand it qualifies as prior art to the ’094 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 GlobalPlatform’s “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.

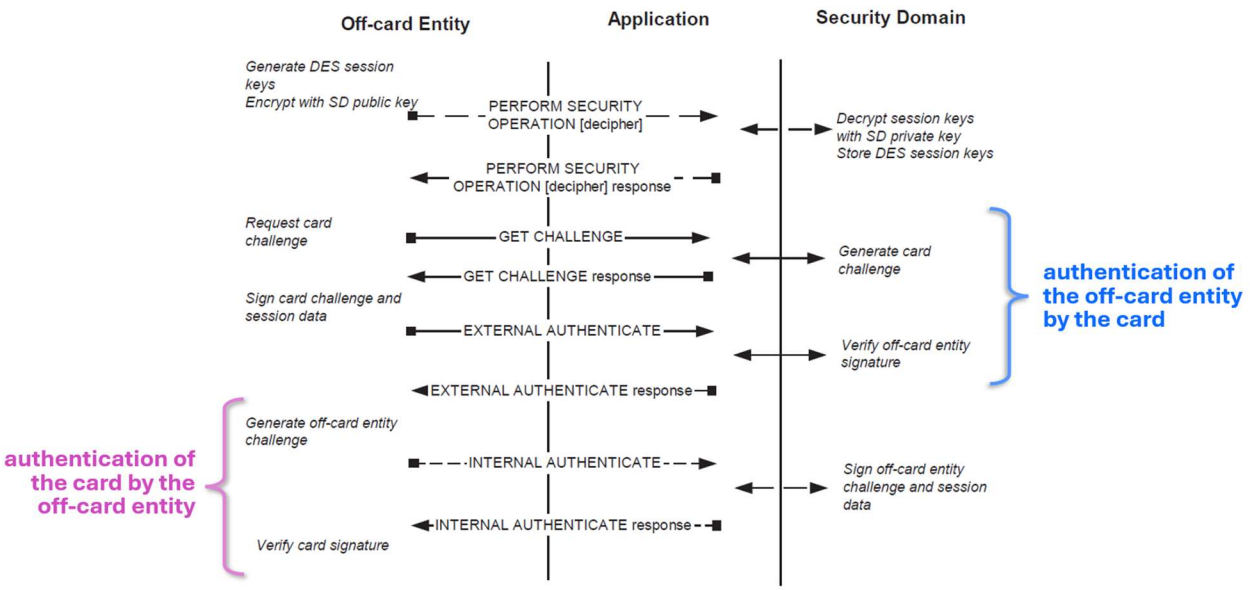


Figure F-9: Entity Authentication Flow

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.

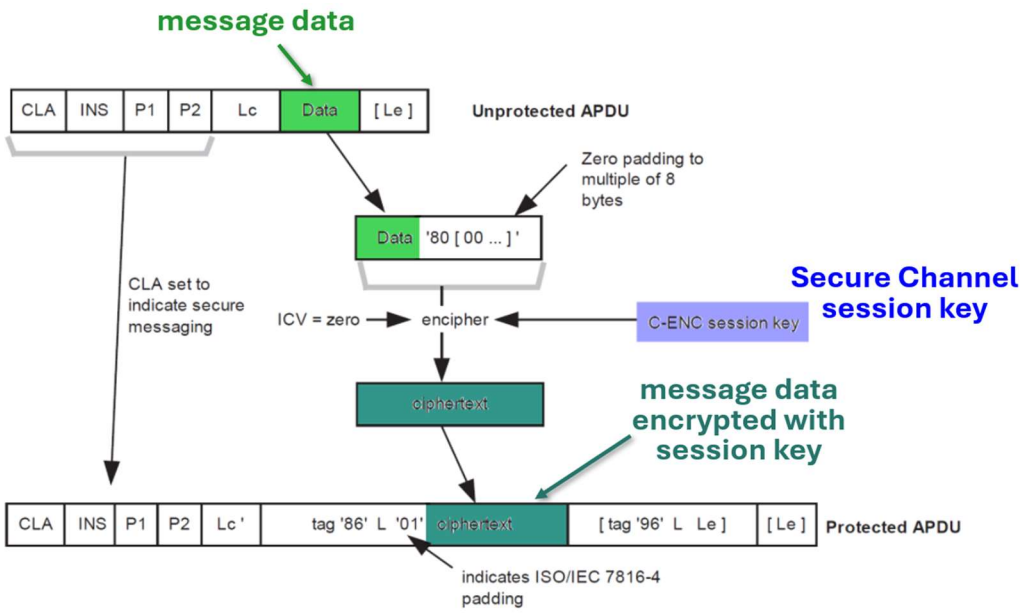


Figure F-11: Secure Messaging: Command message protected for confidentiality

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 (C-ENC), 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 '094 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:

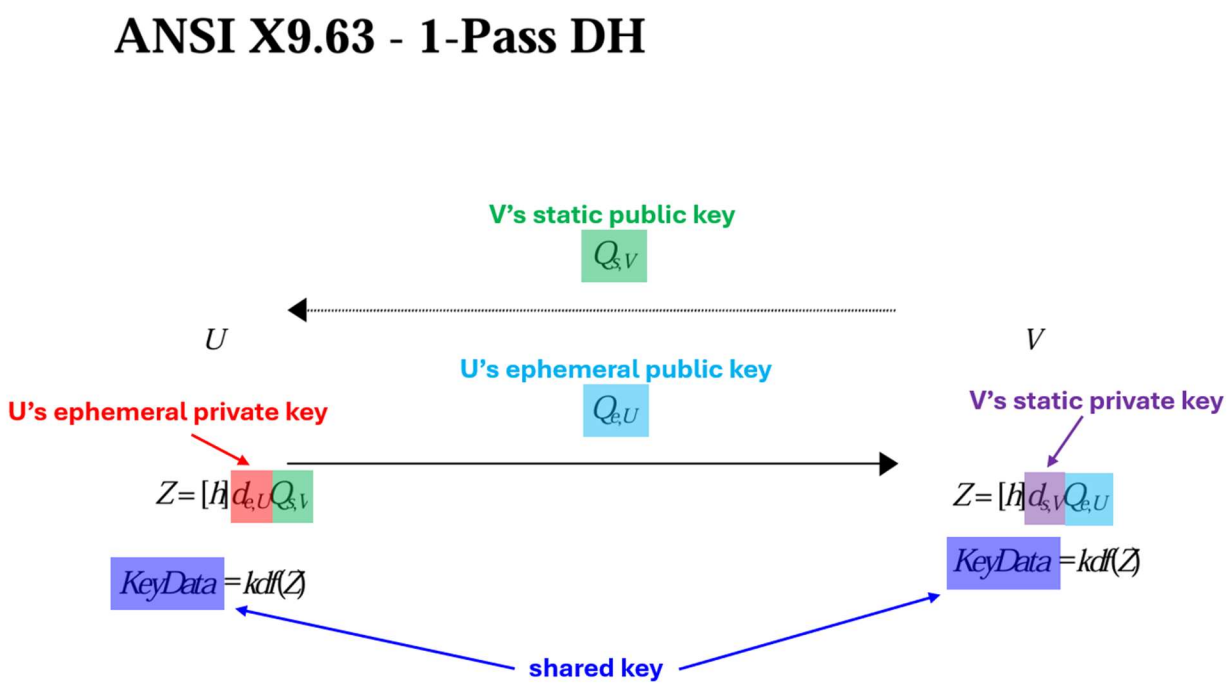
> Alice selects a random number n_a and computes $Y_a = n_a B$, she sends Y_a to Bob. Similarly Bob computes $Y_b = n_b B$ and sends Y_b to Alice. Alice and Bob generate the same key $K = n_a Y_b = n_b Y_a = n_a n_b B$.

Ex-1007, 3. The derived key K in AbiChar corresponds to the profile key derived in Claim 1 of the '094 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 '094 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:



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. Weiss (Ex-1009)

108. Weiss (U.S. Patent Publication No. 2015/0350881) filed on December 17, 2013, as PCT Application No. PCT/EP2013/003.813, claiming foreign application priority to December 21, 2012. I understand it qualifies as prior art to the '094 Patent.

109. Weiss teaches that "secure elements, such as SIMs, eUICCs, UICCs" (Ex-1009, ¶3) may be included in a variety of devices, "such as a tablet or notebook computer, a TV system, a set top box, a vending machine, a vehicle, a surveillance camera, a sensor device and the like" (*id.*, ¶31).

F. Nix175 (Ex-1016)

110. 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.**, I understand that Nix175 is prior art to the '094 Patent. Nix175 shares a specification with the challenged patent.

G. SGP.22 (Ex-1010)

111. SGP.22 (RSP Technical Specification Version 2.0) published in October, 2016. Because the ’094 Patent is entitled to a priority date no earlier than the filing date of the ’893 Patent in its priority chain, as explained in Section IX, I understand that SGP.22 qualifies as prior art to the ’094 Patent.

112. SGP.22 defines procedures for remotely provisioning operator profiles—including Over-The-Air (OTA) keys—from a Subscription Manager (SM-DP+) to an eUICC. Ex-1010, 7. An Elliptic Curve Key Agreement procedure is used to establish a session key (S-ENC) used to protect the profile during transmission to the eUICC. Ex-1010, 36. SGP.22 teaches two ways for using S-ENC—(1) encrypting the profile directly with S-ENC, or (2) encrypting the profile with a random per-profile symmetric key (PPK-ENC), encrypting PPK-ENC with S-ENC, and then transmitting both to the eUICC. Ex-1010, 27-28.

H. SCP11 (Ex-1011)

113. SCP11 (GlobalPlatform Card Secure Channel Protocol ’11’ Card Specification v2.2 - Amendment F) published in May 2015. Ex-1018. Because the ’094 Patent is entitled to a priority date no earlier than the filing date of the ’893 Patent in its priority chain, as explained in Section IX, I understand that SCP11 qualifies as prior art to the ’094 Patent.

114. The SCP11 protocol uses session keys (S-ENC, S-MAC) derived through an elliptic curve key agreement between an off-card entity and a smart card. Ex-1011, 12. SCP11 also uses a Data Encryption Key (DEK)—either a static "Key-DEK" or a session-specific "S-DEK"—to provide an additional layer of protection for sensitive data included within the messages. Ex-1011, 20-21.

XII. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS

A. Ground 1: Claims 1-17 and 19-22 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.

115. A POSITA would have been motivated to combine Park's secure profile provisioning architecture (SPA) with GlobalPlatform's procedures for mutual authentication and 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.

116. Park's Secure Profile Provisioning Architecture (SPA) defines an architectural framework for secure eUICC profile provisioning but intentionally leaves some implementation-level details to the practitioner. Ex-1005, 299-303. Park

generates profiles and securely delivers them from the SM-DP to the eUICC, noting "diverse key agreement protocols, secure communication protocols and cryptographic algorithms can be applied depending on the circumstances around the eUICC." *Id.*, 303. Moreover, Park recognizes the GlobalPlatform Card Specification (GP) and its Secure Channel Protocol (SCP) as the "de facto standard for profile provisioning," and presents its SPA as extending GlobalPlatform. Ex-1005, 298. Park's SPA's fundamental security requirements—mutual authentication, and profile confidentiality—are derived directly from GlobalPlatform. Park explains that SPA's security requirements "should be guaranteed in accordance with the requirements in [5]," where reference [5] corresponds to GlobalPlatform. *Id.*, 298, 303.

117. Thus, a POSITA would have looked to GlobalPlatform to understand how to implement Park's security requirements. GlobalPlatform's SCP's were well-known, widely deployed mechanisms for providing mutual authentication and confidentiality in smart-card provisioning. Incorporating GlobalPlatform's authentication and 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.

118. **<u>Mutual Authentication:</u>** The combined Park-GlobalPlatform system would achieve mutual authentication between the eUICC and the SM-DP using random challenges. GlobalPlatform's SCP10 performs PKI authentication whereby each side sends a random challenge to the other side, which then signs it with its private key. Ex-1006, 246-47, 266-68, 271. The sender then verifies that private key signature using the responder's public key, as illustrated in Figure F-9. *Id.*

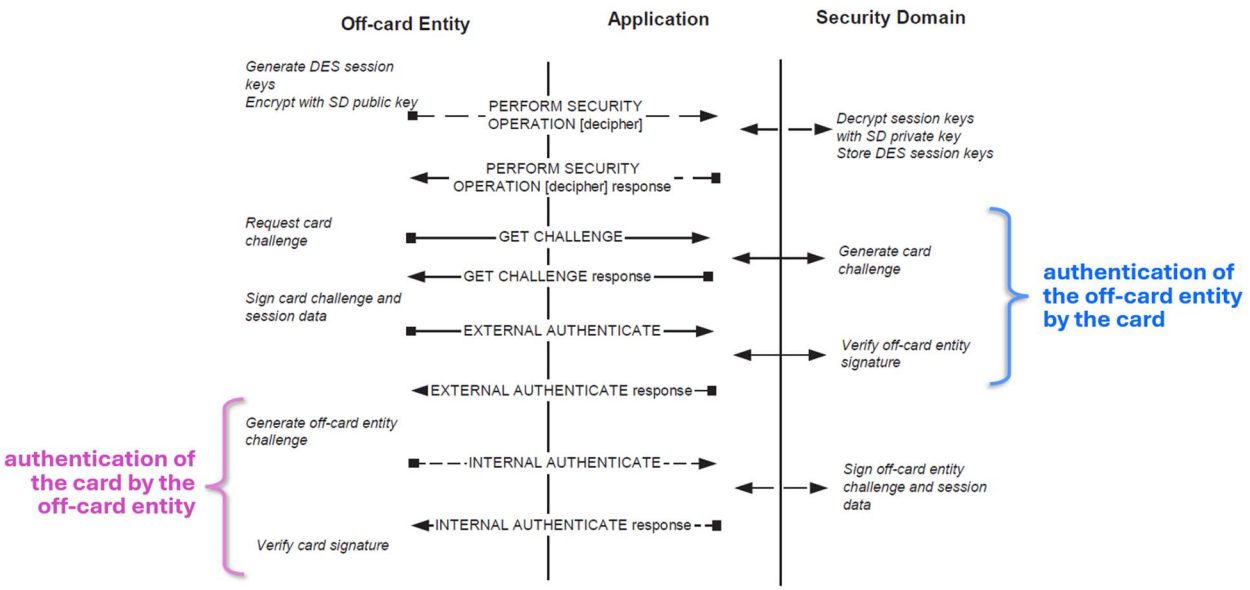
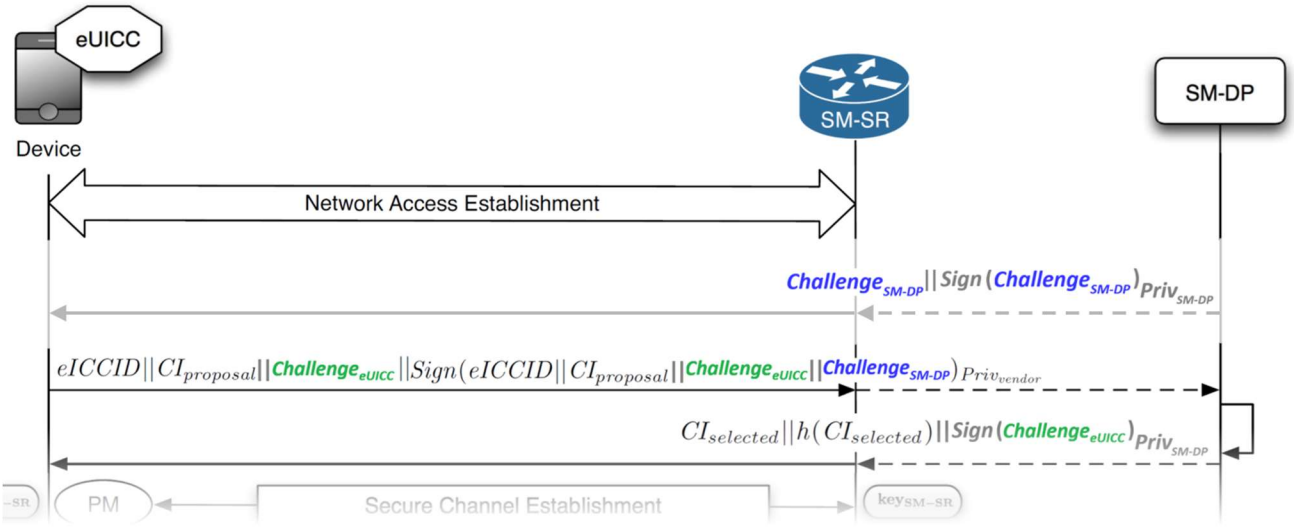


Figure F-9: Entity Authentication Flow

119. Given the limited processing and communication resources of eUICCs, a POSITA would have found it advantageous to include these random challenges and responses (to the extent possible) in existing messages sent between the eUICC to the SM-DP—such as Park's capability negotiation message and the SM-DP's response—rather than in a new communication. As illustrated in the modified

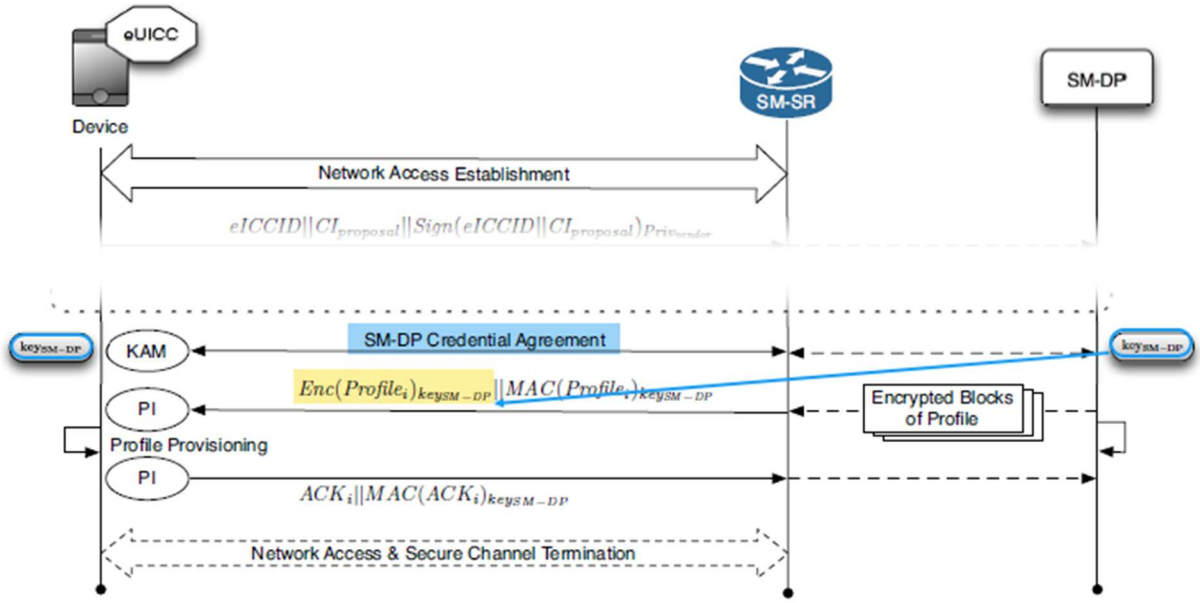
version of Park's Figure 4, below, it would have been a straightforward process to add these random challenges and responses to Park's capability negotiation messages:



A POSITA would also have recognized that doing so would have had the added benefit of adding replay protection for the capability negotiation message, ensuring that an intercepted or previously transmitted message could not be reused by an attacker.

120. **<u>Confidentiality:</u>** In Park's SPA, the eUICC's Key Agreement Module (KAM) is used to generate a shared SM-DP (below, blue) credential for encrypting and decrypting the profile (below, yellow) transmitted from the SM-DP to the eUICC. Ex-1005, 302. In the combined Park-GlobalPlatform system, this SM-DP credential 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.



121. As Park explains, the SM-DP uses the SM-DP credential, together with "selected cryptographic algorithms" agreed upon by the parties during the capability negotiation process, to encrypt the profile that it sends to the eUICC. Ex-1005, 300. GlobalPlatform provides those established encryption procedures.

122. As explained in Section XII.A.1 (motivation to combine), when transmitting sensitive data, such as secret/private keys, SPC10 first encrypts the sensitive data "us[ing] the relevant data encryption session key (DEK)" and then further encrypts the message with the secure channel session key, providing layered protection for these highly sensitive materials. *Id.*, 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.

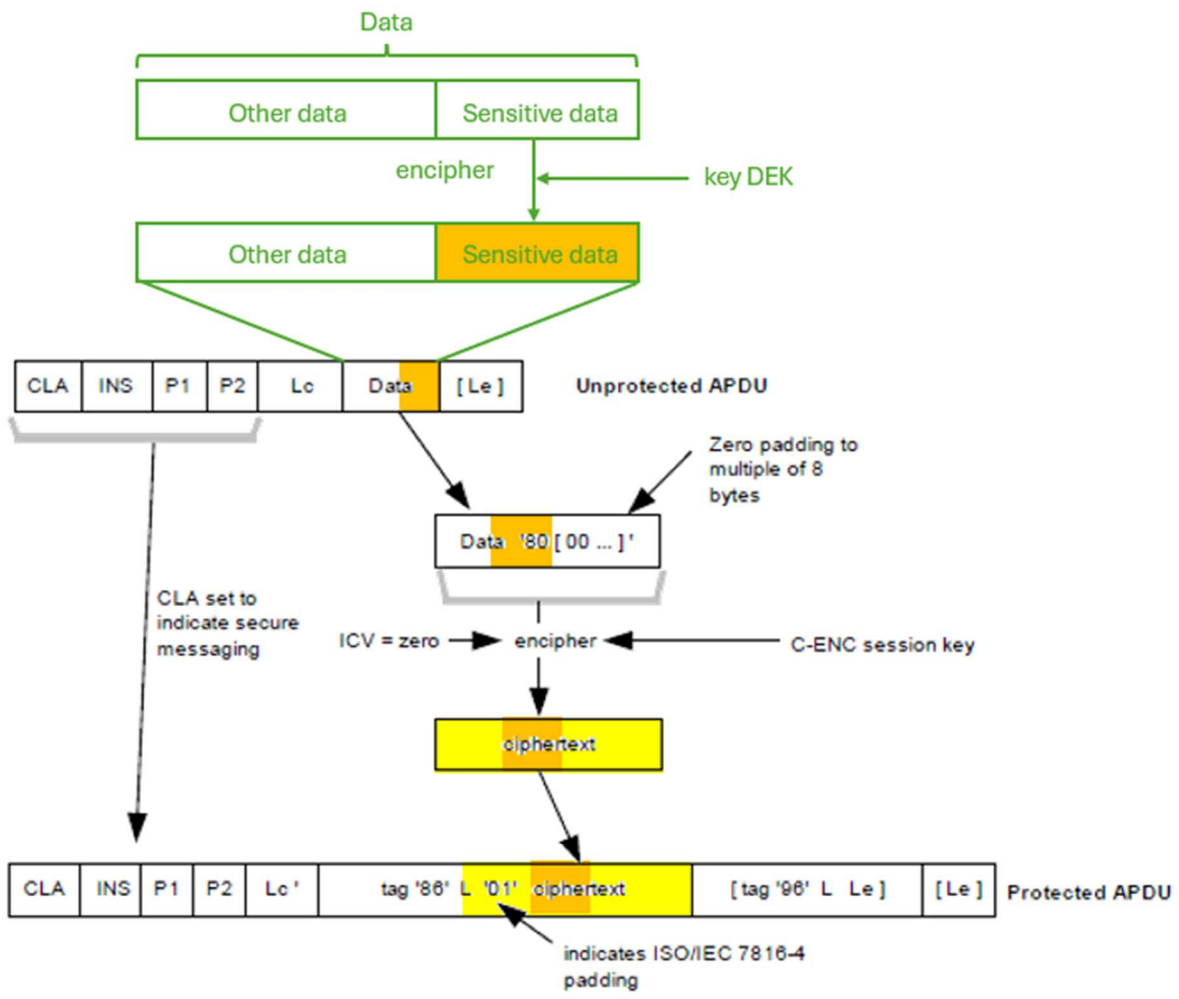


Figure F-11: Secure Messaging: Command message protected for confidentiality

123. 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, Park-GlobalPlatform sends profiles from the SM-DP to the eUICC that include an inner layer of encryption for sensitive materials like keys (using GlobalPlatform's DEK), as well as an outer layer of encryption using Park's $key_{SM\text{-}DP}$ (corresponding to GlobalPlatform's C-ENC session key).

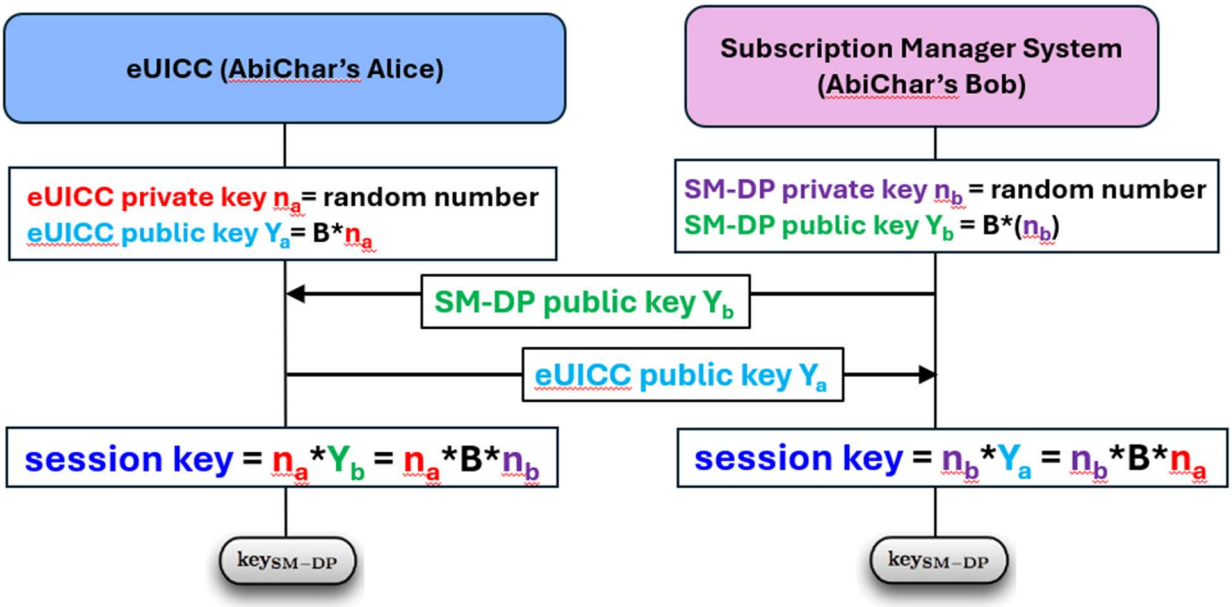
b. A POSITA would have understood that Abi-Char's elliptic curve Diffie-Hellman key agreement could be implemented within Park's SPA to provide an efficient asymmetric key-exchange mechanism

124. Park's Key Agreement Module (KAM) "provides the ***asymmetric algorithm-based key agreement protocol***" used to derive $key_{SM\text{-}DP}$ for encrypting the eUICC profile. Ex-1005, 301. But Park does not specify a particular algorithm, instead explaining that "diverse key agreement protocols … can be applied depending on the circumstances around the eUICC." Ex-1005, 303.

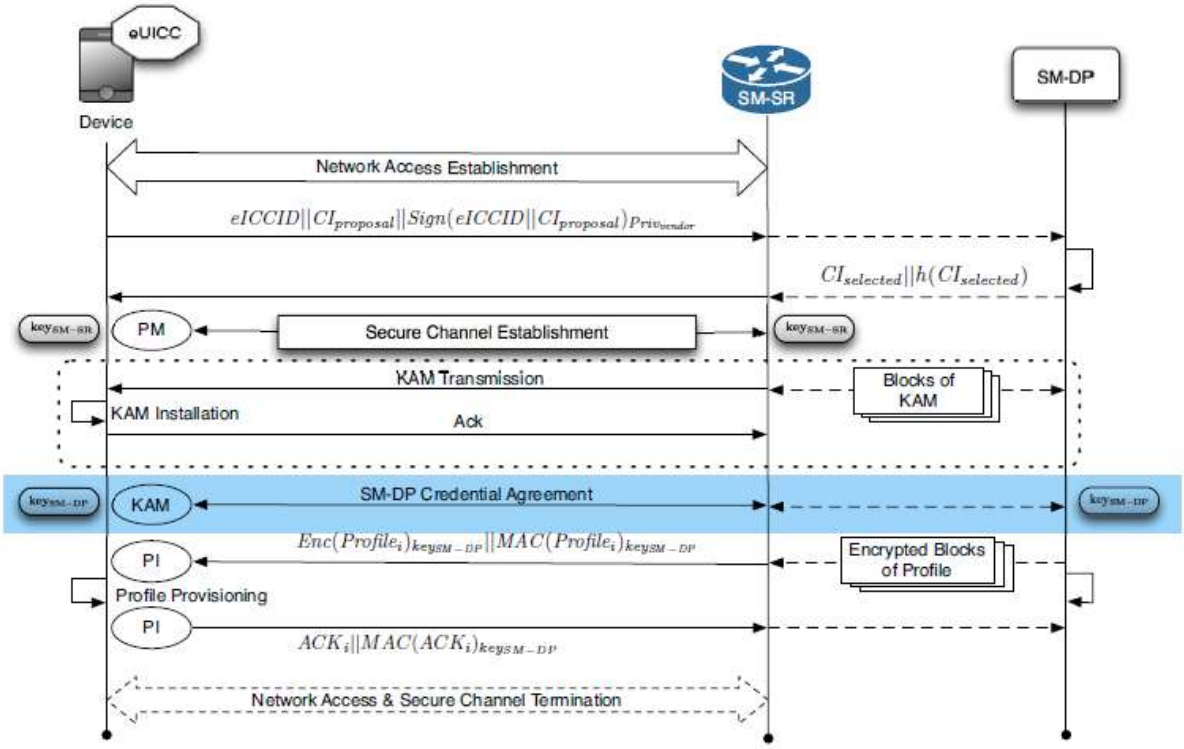
125. A POSITA would have recognized that the elliptic-curve-Diffie-Hellman key agreement protocol described in AbiChar provides such an efficient

asymmetric key agreement technique suitable for Park's architecture. Abi-Char explains 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 describes an elliptic-curve Diffie-Hellman (ECDH) key-agreement exchange, wherein each party generates random private keys (n_a and n_b respectively), generates public 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.*

126. A POSITA would have incorporated AbiChar's ECDH protocol within Park's flexible, asymmetric framework to provide a secure, lightweight, and computationally efficient mechanism for establishing $key_{SM\text{-}DP}$ 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 is illustrated as follows:



127. 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 keySM-DP in the "SM-DP Credential Agreement" step:



2. Independent Claim 1

a. Element 1[pre]: A method comprising:

128. Park, GlobalPlatform, and AbiChar ("Park-GlobalPlatform-AbiChar") teaches the preamble, to the extent it is limiting.

129. Park teaches a method for provisioning a profile onto an "[e]mbedded UICC (hereafter eUICC)," using a "secure profile provisioning architecture." Ex-1002, 297.

b. Element 1[a][1]: (a) generating a first message comprising

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

131. Park teaches that the eUICC and SM-DP negotiate to establish the key exchange protocol to be used by the parties. Ex-1005, 301. The eUICC sends a message to the SM-SR that includes "the eUICC ID (*eICCID*) and its capability information ($CI_{proposal}$)," as illustrated in Figure 4, below. Ex-1005, 300. This capability negotiation message corresponds to the claimed "first message."

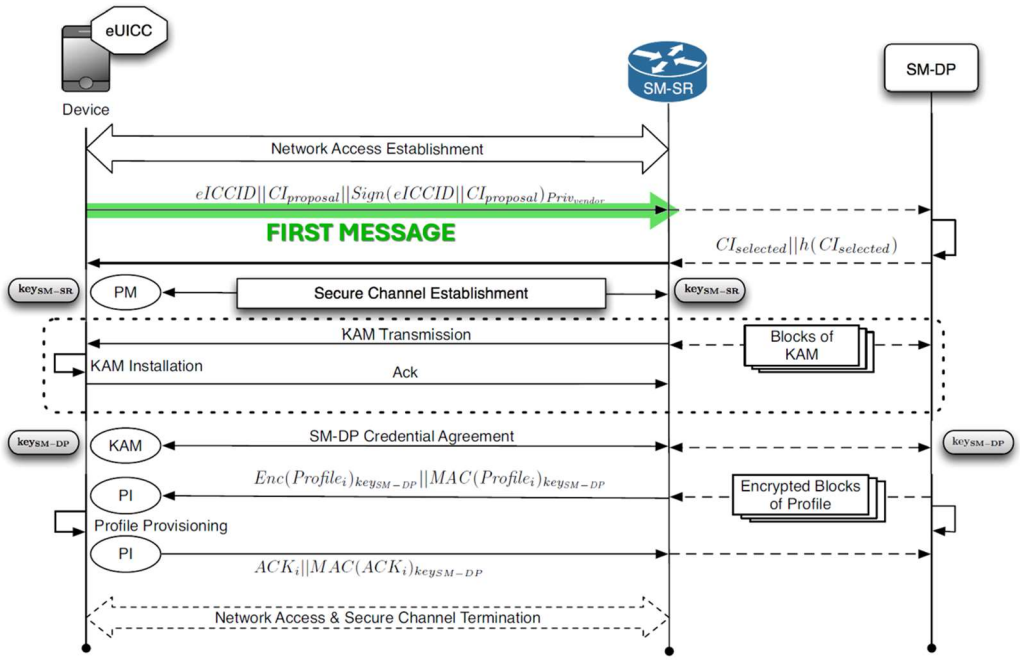
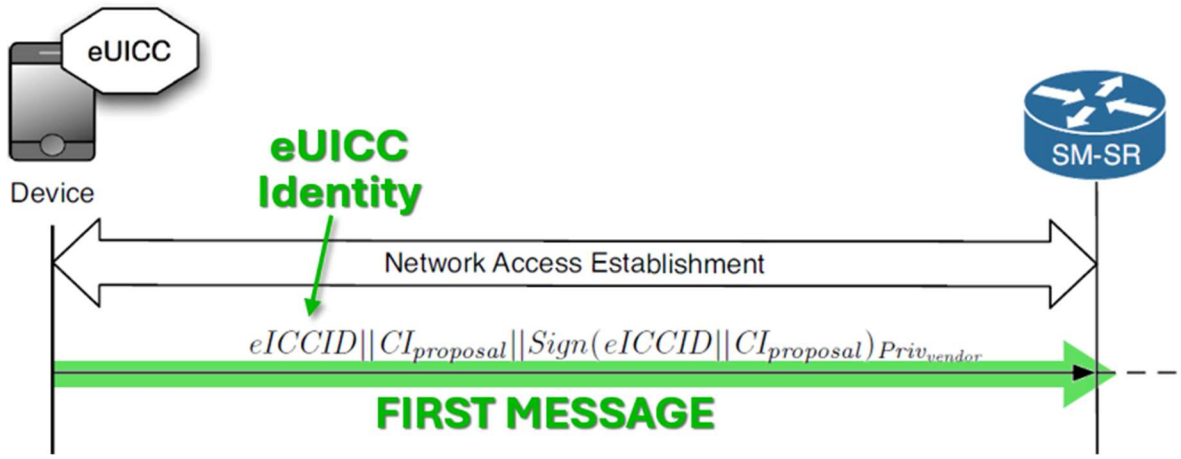


Fig. 4. Secure Profile Provisioning Procedure

c. Element 1[a][2][i]: (1) an identity of an embedded universal integrated circuit card (eUICC);

132. Park-GlobalPlatform-AbiChar teaches Element 1[a][2][i]. As illustrated in Figure 4, excerpted below, Park's capability negotiation message (the claimed "first message") includes an "eUICC ID (*eICCID*)." Ex-1005, 300.



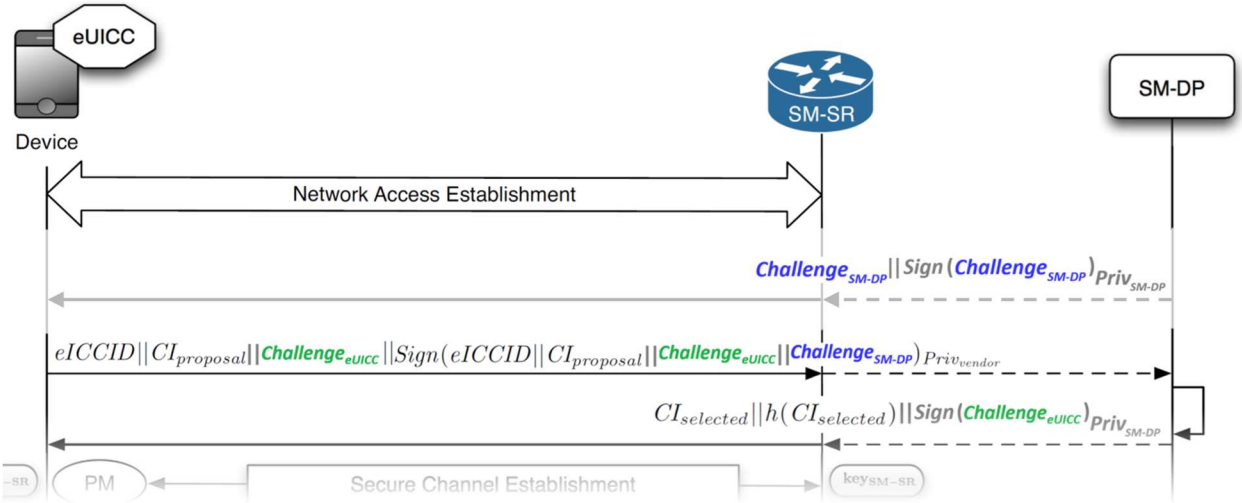
Ex-1005, Fig. 4 (excerpted)

133. Park explains that the "eUICC ID" is an "[i]dentification for eUICC to identify a particular eUICC." *Id.*, 299. Thus, the eUICC ID within Park's capability negotiation message (claimed "first message") corresponds to the claimed "identity of an embedded universal integrated circuit card (eUICC)."

d. Element 1[a][2][ii]: (2) a nonce; and

134. Park-GlobalPlatform-AbiChar teaches Element 1[a][2][ii].

135. As explained in Section XII.A.1 (motivation to combine), Park-GlobalPlatform-AbiChar would use GlobalPlatform's PKI authentication scheme for authentication. This scheme involves issuing random challenges and verifying private key signatures applied to those challenges (Ex-1006, 246-47, 266-68, 271), and would have been applied to Park's capability negotiation procedure, as illustrated in the modified version of Park's Figure 4, below:



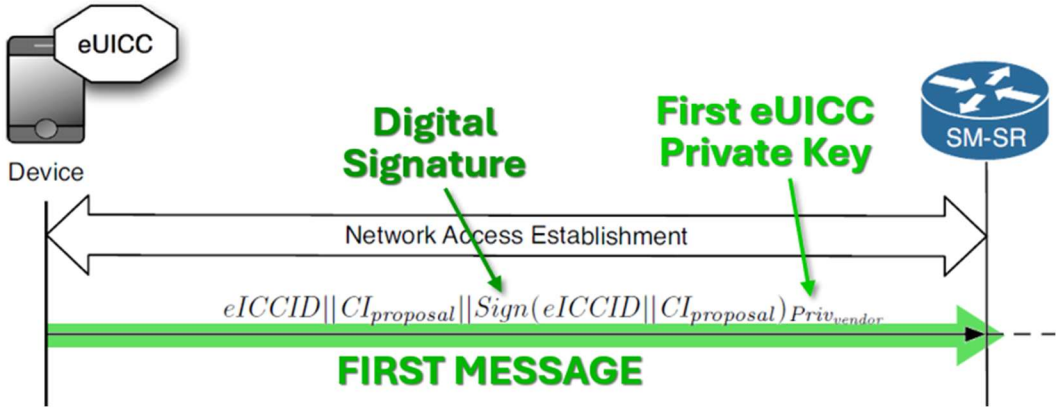
136. Thus the SM-DP first issues a random challenge (*Challenge*$_{SM\text{-}DP}$, blue) to the eUICC. Then, the eUICC responds, by signing the SM-DP's challenge with its private key (*Sign*(…*Challenge*$_{SM\text{-}DP}$)*Priv*$_{vendor}$) and issuing its own random challenge to the SM-DP (*Challenge*$_{eUICC}$, green)—both within the initial capability negotiation message, for efficiency purposes. A POSITA would have understood that each random challenge is a random number intended to be used only once—i.e., a nonce. Thus, the eUICC's capability negotiation message (the claimed "first message") includes two different nonces, as illustrated below—(1) the random challenge (*Challenge*$_{eUICC}$, green) issued to the SM-DP, and (2) the SM-DP's random challenge that has been signed with the eUICC's private key (*Sign*(…*Challenge*$_{SM\text{-}DP}$)*Priv*$_{vendor}$) for subsequent use by the SM-DP in authenticating the eUICC.

$$eICCID||CI_{proposal}||Challenge_{eUICC}||Sign(eICCID||CI_{proposal}||Challenge_{eUICC}||Challenge_{SM\text{-}DP})_{Priv_{vendor}}$$

each a "nonce"

e. Element 1[a][2][iii]: (3) a first digital signature, generated using a first eUICC private key, wherein the first eUICC private key corresponds to a first eUICC public key;

137. Park-GlobalPlatform-AbiChar teaches Element 1[a][2][iii]. Park explains that, "[t]he eUICC ID and capability information are signed by the eUICC Vendors private key," $Priv_{vendor}$ and this signed value, Sign(…)$_{Privvendor}$ is added to the message, as illustrated in excerpted Figure 4, below. Ex-1005, 300.

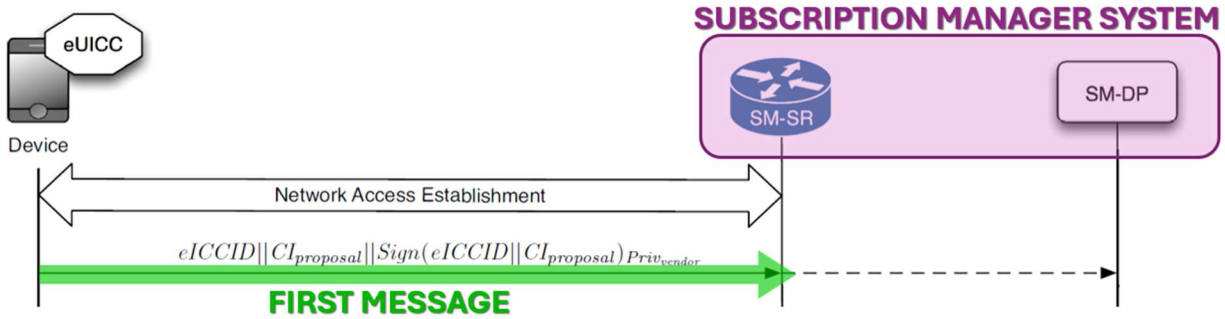


138. Park explains that the eUICC Vendor private key is part of a public key pair (i.e., it corresponds to an eUICC Vendor public key), and that it is generated by the eUICC Vendor and stored on the eUICC during the pre-provisioning stage. Ex-1005, 300.

f. Element 1[b]: (b) sending the first message via a network application to a subscription manager system;

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

140. Park teaches sending the capability negotiation message (claimed "first message") from the eUICC to the SM-DP through the secure channel provided by the SM-SR, as illustrated in Figure 4, excerpted below. The SM-DP and SM-SR correspond to distinct logical and/or physical entities within Park's Subscription Manager (SM).



141. As illustrated in Figure 5, below, messages sent from the eUICC to the SM are sent using a Device Module (DM), which is "software running in the Device to perform the roles of the intermediator between eUICC and SM-SR." Ex-1005, 299. Park's DM corresponds to the claimed "network application."

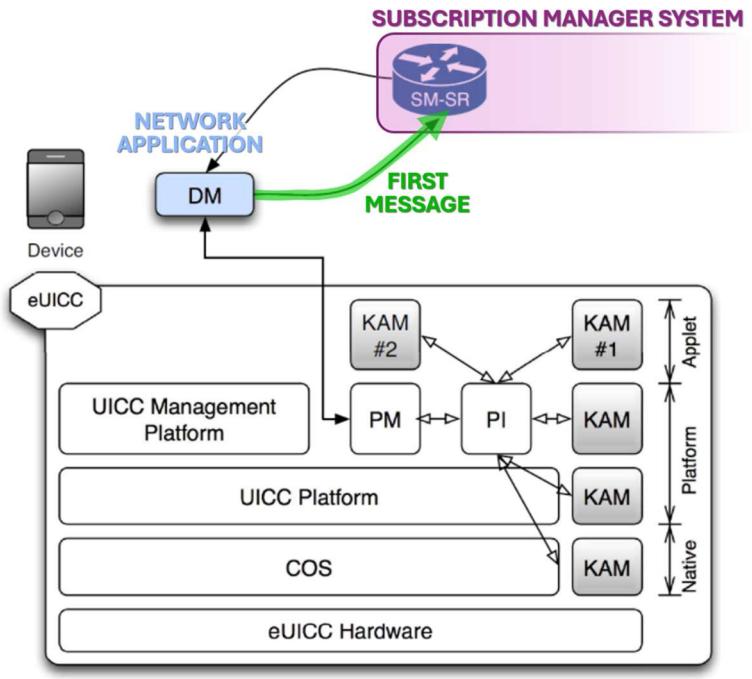
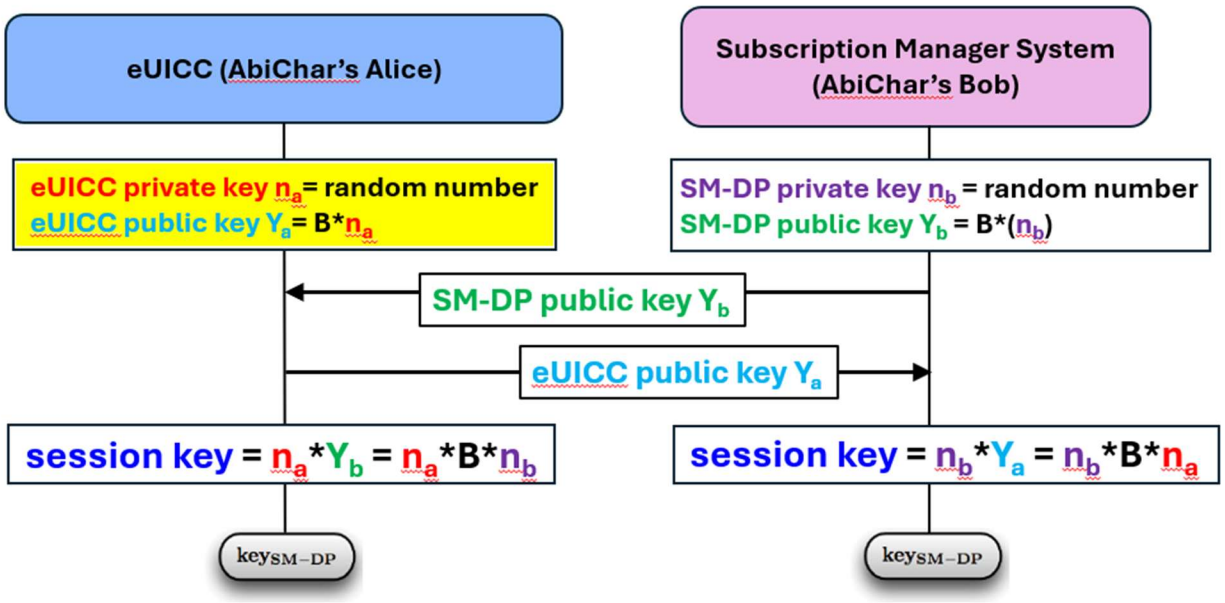


Fig. 5. Internal Architecture of eUICC

g. Element 1[c]: (c) deriving a second eUICC private key and a corresponding second eUICC public key using a first random number generator and a first set of cryptographic algorithms;

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

143. As explained in Section XII.A.1 (motivation to combine), the combined system uses AbiChar's ECDH key agreement protocol illustrated below:



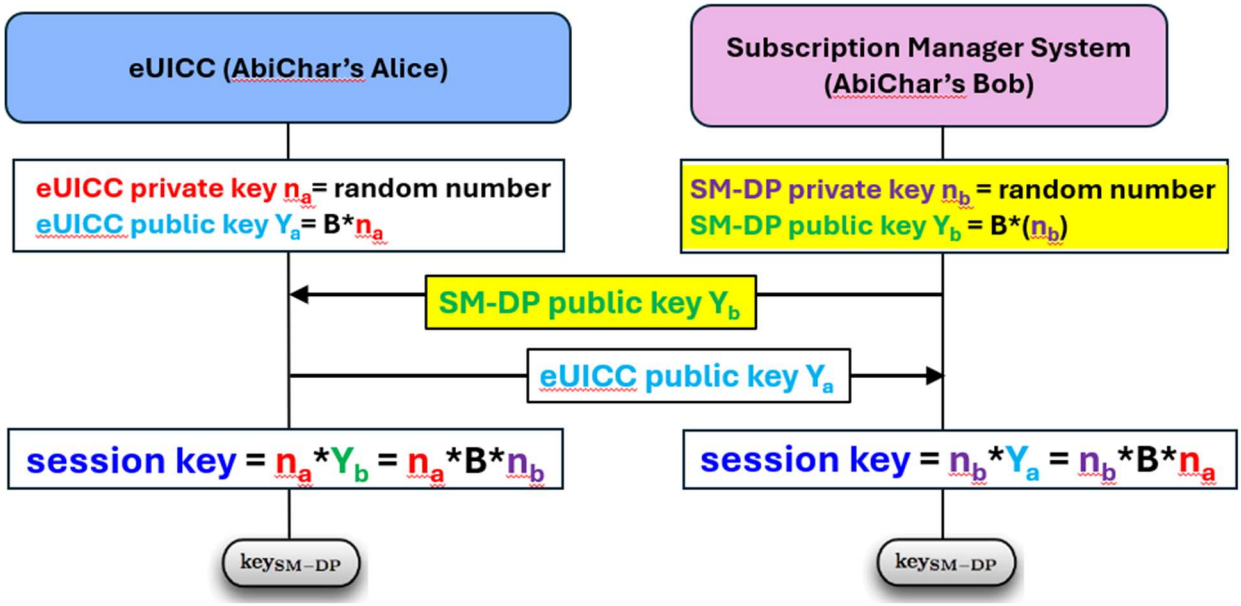
144. The eUICC's **private key n_a** (claimed "second eUICC private key") is generating using a random number. Ex-1007, 3. The eUICC calculates corresponding **public key $Y_a = B*n_a$** (claimed "second eUICC public key") from **private key n_a**. Ex-1007, 3. The value B used to calculate the eUICC public key Y_a is a point on an elliptic curve. Ex-1007, 3. Thus, the eUICC public key (Y_a) is calculated from the randomly-generated eUICC private key (n_a), using elliptic-curve scalar multiplication—the fundamental cryptographic operation of ECDH.

h. Element 1[d]: (d) storing a subscription manager public key which corresponds to a subscription manager private key;

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

146. In AbiChar's ECDH key exchange, Park's SM-DP uses a random number to generate a **private key n_b** (claimed "subscription manager private key"),

and then calculates **a corresponding public key** $Y_b = B*n_b$ (claimed "subscription manager public key") from that private key n_b. Ex-1007, 3.



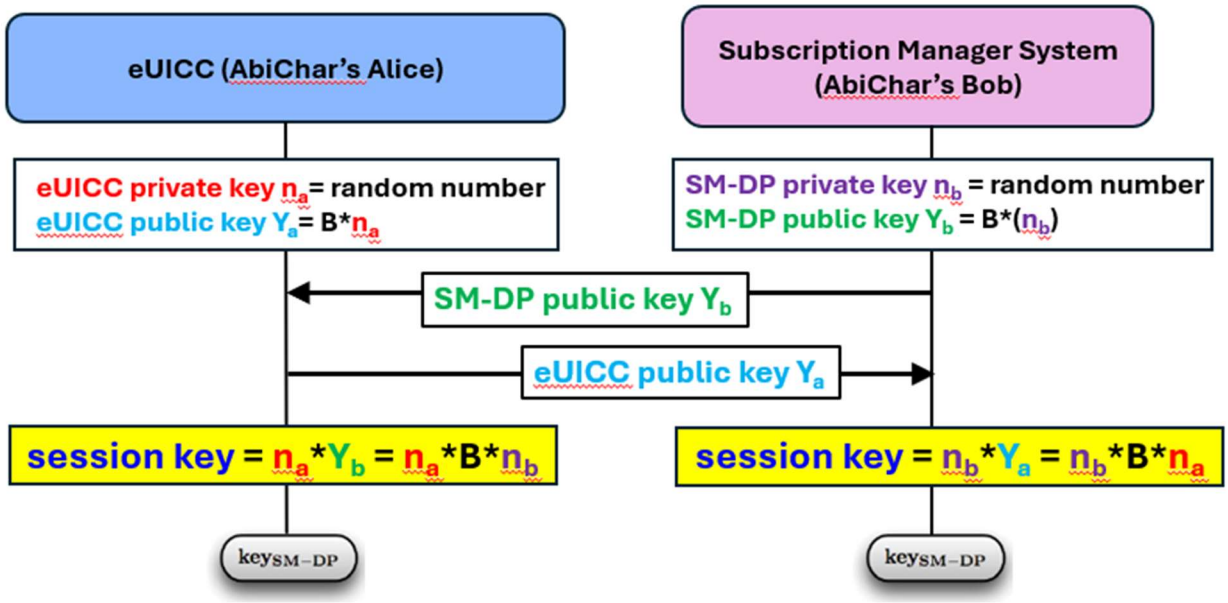
147. The SM-DP then sends its **public key** Y_b (claimed "subscription manager public key") for use by the eUICC in calculating the shared **session key** (SM-DP Credential). Ex-1007, 3; Ex-1005, 301. A POSITA would have understood that the eUICC would store the **public key** Y_b (claimed "subscription manager public key) at least long enough for the eUICC to use it in the key exchange algorithm.

i. Element 1[e][1]: (e) deriving a profile key using a key exchange algorithm based on at least:

(i) the second eUICC private key, and

(ii) the stored subscription manager public key,

148. Park-GlobalPlatform-AbiChar teaches Element 1[e][1].



149. As illustrated above, the eUICC calculates the **shared SM-DP Credential/session key** (claimed "profile key") using its **private key** n_a (claimed "second 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, 3.

150. Similarly, the SM-DP calculates the **shared session key** using its **private key** n_b (claimed "subscription manager private key"), together with the eUICC **public key** Y_a (claimed "second eUICC public key) that it received from the eUICC. Ex-1007, 3.

j. Element 1[e][2]: wherein the profile key can also be derived at the subscription manager system based at least on: the second eUICC public key, and the subscription manager private key;

151. Similarly, the SM-DP calculates the **shared session key** using its **private key** n_b (claimed "subscription manager private key"), together with the eUICC **public key** Y_a (claimed "second eUICC public key) that it received from the eUICC. Ex-1007, 3.

152. Thus, Park-GlobalPlatform-AbiChar teaches Element 1[e][2].

k. Element 1[f]: receiving from the subscription manager system, an encrypted profile comprising a ciphertext including a key K encrypted with a symmetric key;

153. Park-GlobalPlatform-AbiChar teaches Element 1[f].

154. Park's SM-DP, encrypts the profile "[u]sing the agreed SM-DP Credentials and the selected cryptographic algorithms," $Enc(Profile_i)_{keySM\text{-}DP}$, and then "sends the protected profile blocks to the eUICC." Ex-1005, 300. Park teaches that profiles contain "security-sensitive information," including "keys" used for accessing a mobile network. 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 (using DEK), and then encrypting it

again using the session key, as illustrated below, using GlobalPlatform's Figure F-11. Ex-1006, 258, 260, 264.

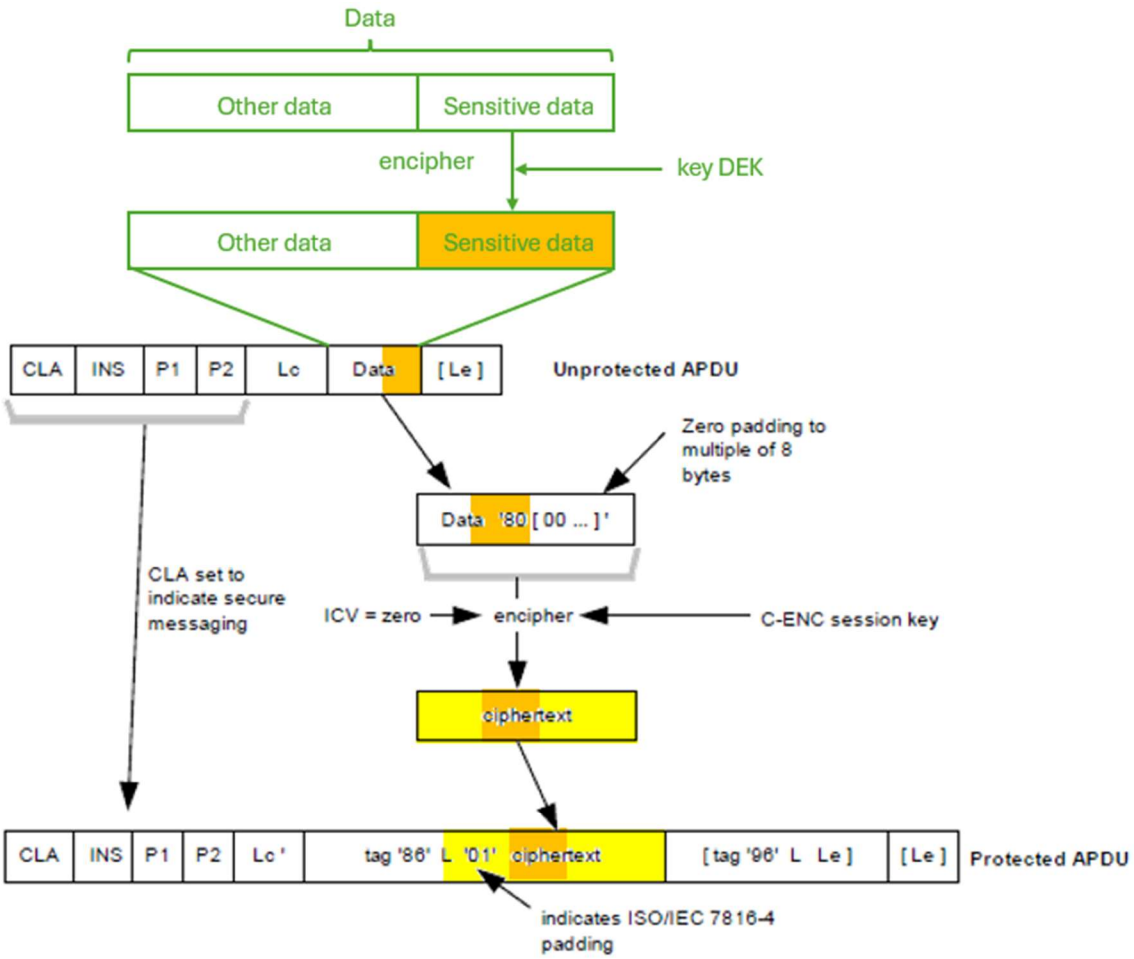


Figure F-11: Secure Messaging: Command message protected for confidentiality

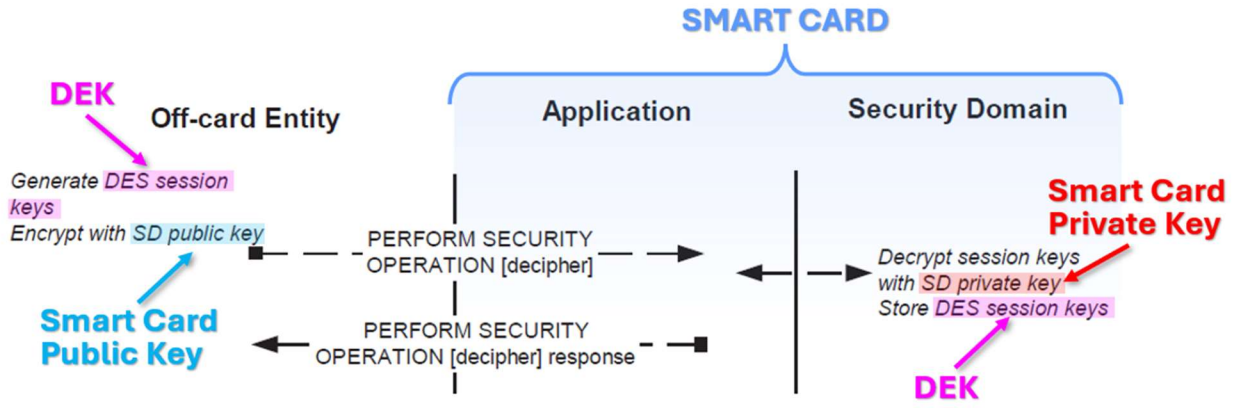
155. GlobalPlatform's DEK-based inner encryption applied to the sensitive keys included within Park's profile adds an additional layer of protection for the most sensitive data within the profile.

156. GlobalPlatform explains that encryption with the DEK "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*** would be used to encrypt a key K included within Park's profile.

l. Element 1[g]: (g) receiving the symmetric key;

157. Park-GlobalPlatform-AbiChar teaches Element 1[g].

158. 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, 247, 275-76. This is illustrated in Figure F-9, excerpted below:



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

159. 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. Further, the SM-DP is part of the network trust zone separate from the SM-SR.

m. Element 1[h]: (h) decrypting, by the embedded universal integrated circuit card, at least a portion of the encrypted profile using the profile key;

160. Park-GlobalPlatform-AbiChar teaches Element 1[h].

161. Park teaches that, upon receiving the encrypted profile block 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 each of the SM-DP and eUICC share the

same **SM-DP Credential** $key_{SM\text{-}DP}$ (claimed "profile key")[3], 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. For clarification, Park's SM-DP credential, or $key_{SM\text{-}DP}$, is the equivalent of GlobalPlatform's C-ENC session key, which it uses to protect the APDU data. Ex-1006, 258-64, Fig. F-11. Thus, a POSITA would understood that decrypting each profile block, as performed by the eUICC, would involve using the **SM-DP Credential** $key_{SM\text{-}DP}$ (claimed "profile key").

n. Element 1[i]: (i) decrypting, by the embedded universal integrated circuit card, at least a portion of the ciphertext using the symmetric key; and

162. Park-GlobalPlatform-AbiChar teaches Element 1[c][9].

163. As described in Section XII.A.1 (motivation to combine), in the combined Park-GlobalPlatform system, certain key materials within the profile (e.g., authentication keys K) are encrypted under a DEK (the claimed "symmetric key") before the entire profile is wrapped under the SM-DP Credential/session key. GlobalPlatform teaches that, in response to receiving such encrypted "sensitive data," the smart card will perform a "decryption" process that "is the exact opposite" of the encryption process. Ex-1006, 264. Thus, once the eUICC has decrypted the

[3] Park's SM-DP credential, or $key_{SM\text{-}DP}$, is the equivalent of GlobalPlatform's C-ENC session key, which it uses to protect the APDU data. Ex-1006, 258-64, Fig. F-11.

outer profile layer using the SM-DP Credential/session key, it would use the received DEK to decrypt the inner ciphertext containing the key K.

o. Element 1[j]: (j) storing at least the key K in the embedded universal integrated circuit card for use in future communications.

164. Park-GlobalPlatform-AbiChar teaches Element 1[j].

165. Park teaches that, following successful decryption of the profile blocks, "the profile blocks are provisioned inside the eUICC." Ex-1005, 300. A POSITA would have understood that this provisioning process would involve storing the key K in the eUICC, so that it can be used by the eUICC to access the mobile network corresponding to the provisioned profile. Ex-1005, 297 (MNO profile includes "keys").

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

3. Dependent Claims 2-17, 19-22

a. Claim 2: The method of claim 1, wherein the subscription manager public key is downloaded via the network application.

167. Park-GlobalPlatform-AbiChar teaches Claim 2.

168. As explained for Element 1[d], during the key exchange between the eUICC and the SM-DP, the SM-DP sends the eUICC its **public key Y_b** (claimed "subscription manager public key") for use by the eUICC in calculating the shared session key (SM-DP Credential). Ex-1007, 3; Ex-1005, 301. Park's Figure 5, below,

shows that the SM-DP sends messages to the eUICC "via the secure communication channel of SM-SR," and messages sent via the SM-SR to the eUICC are received using the eUICC's Device Module (DM)—i.e., the claimed "network application." Ex-1005, 299-300

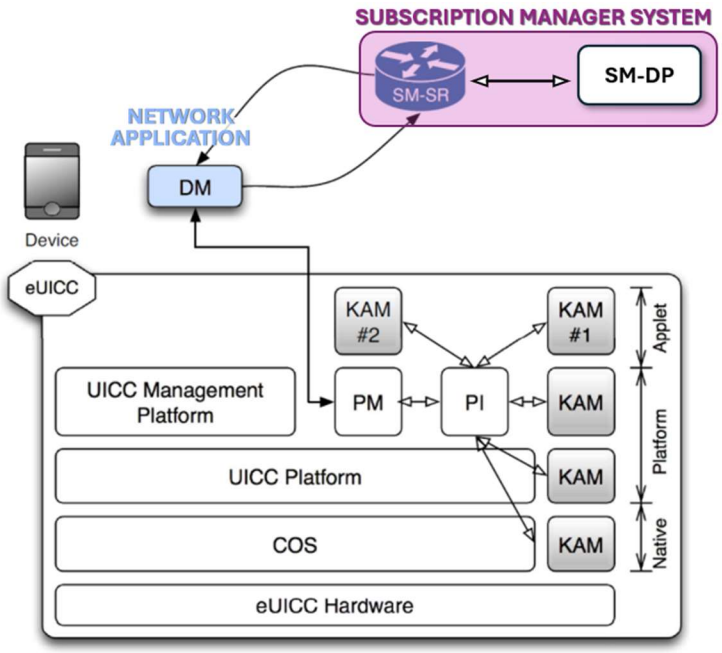


Fig. 5. Internal Architecture of eUICC

Ex-1005, Fig. 5 (modified to illustrated the interworking between the SM-SR and SM-DR).

b. Claim 3: The method of claim 1, wherein the step of storing the subscription manager public key occurs at the time of manufacture of the embedded universal integrated circuit card.

169. Park-GlobalPlatform-AbiChar teaches Claim 3.

170. AbiChar's key-exchange protocol uses ephemeral key pairs. Ex-1006, 3. A POSITA, however, would have recognized that substituting a long-term static key pair for the server-side entity—corresponding here to Park's SM-DP—could provide practical advantages in the context of Park's secure profile-provisioning architecture. Park characterizes the eUICC as a "resource-constrain[ed]" device for which "overhead for the security-related operations" should be minimized. Ex-1005, 299. Thus, when the eUICC has limited computational and communication resources, it may be advantageous to simplify the key-exchange procedure by using a static key pair on the SM-DP side, thereby reducing the messaging required to establish the shared secret.

171. In such a modified ECDH scheme, the eUICC would use the **SM-DP static public key** to calculate the shared SM-DP Credential/session key prior to engaging in the key exchange, in the same manner as described in Section XII.A.2.i. A POSITA would have recognized that a natural way to provide the SM-DP public key to the eUICC would be "during the eUICC-based device manufacturing," as part of Park's "Pre-Provisioning Procedure." Ex-1005, 300. In particular, a POSITA would have recognized that it would be straightforward to add the SM-DP's private key to "the SM-SR Credentials (key_{SM-SR}) and the capability information ($CI_{installed}$) of each eUICC" that "are installed into each eUICC" during the pre-provisioning

procedure. *Id*. Indeed, this would enable the eUICC would use the same **SM-DP static public key** to calculate the shared SM-DP Credential/session key as it would have used prior to engaging in the key exchange, to authenticate the SM-DP (by verifying the SM-DP's private key signature on the challenge it issued to the SM-DP). *See* Section XII.A.2.e.

c. Claim 4: The method of claim 1, wherein the key exchange algorithm is a Diffie-Hellman key exchange.

Claim 5: The method of claim 1, wherein the key exchange algorithm is an Elliptic Curve Diffie-Hellman key exchange.

172. Park-GlobalPlatform-AbiChar teaches Claims 4-5.

173. AbiChar teaches a "Diffie-Hellman key exchange based on EC"—i.e., an Elliptic Curve Diffie-Hellman key exchange. Ex-1007, 3.

d. Claim 6: The method of claim 1, wherein the encrypted profile is encrypted in a manner so that intermediate nodes on an IP network would not be able to read data within the profile in an unencrypted form.

174. Park-GlobalPlatform-AbiChar teaches Claim 6.

175. Park's secure profile provisioning architecture (SPA) provides end-to-end security for a profile sent from the SM-DP to the eUICC. Ex-1005, 297, 301. As Park recognized, because the SM-SR may be operated by an entity distinct from the SM-DP / MNO—e.g., a third-party—it therefore should not be trusted with access to sensitive profile contents. *Id*., 301 (explaining that, for security purposes, "the

SM-DP Credentials should not be revealed to any party, even SM-SR, except for eUICC"). Park's SPA addresses precisely this architectural challenge, defining an overall framework in which confidentiality between the SM-DP and the eUICC is ensured by having the eUICC and SM-DP agree on profile encryption keys directly, without exposing those keys to the SM-SR or any other intermediaries. *Id.*, 299-301. This ensures that no intermediate nodes in the network (including the SM-SR) are able to read data within the profile. *Id*.

e. Claim 7: The method of claim 1, wherein subscription manager system stores the ciphertext and the symmetric key before step (a).

176. Park-GlobalPlatform-AbiChar teaches Claim 7, for the reasons described below.

(1) the subscription manager system stores the ciphertext and the symmetric key

177. Park's SM-DP prepares and delivers profiles enabling an eUICC to access an MNO's mobile network. Ex-1005, 297. A POSITA would have recognized that for security reasons, the MNO-specific network-access credentials, such as key K included within the profiles would not be transmitted or stored in plaintext form when exchanged between the MNO and the SM-DP, or otherwise handled during profile preparation. For example, GlobalPlatform SCP10 recognizes that sensitive data, such as key data, must be encrypted over and above the level of encryption

applied to the transmission channel. E.g., Ex-1006 §F.3.2.7 ("[A]ll keys transmitted to a card (e.g. in a PUT KEY command) should be encrypted. Data encryption is over and beyond the Current Security Level required for the Secure Channel Session."). Similarly, the National Institute of Standards and Technology (NIST) teaches "[s]egregating keys from general data" and encrypting such keys with a Key-Wrapping Key (KWK), to "provide an extra layer of protection" for sensitive key data, as discussed in Background Section VII.A.1.c. Ex-1024, NIST SP 800-38F, at 2. Thus, a POSITA would have understood the need to perform a separate encryption of sensitive data and naturally would have saved such data in encrypted form to enhance security and simplify data transmission because such data would already be stored in encrypted form. Indeed, this sensitive data (including key K) would be protected in ciphertext form—that is, encrypted under a symmetric data-encryption or key-wrapping key (such as DEK))—to ensure the secure transfer of the key material.

178. A POSITA would further have recognized that, once the SM-DP had obtained the encrypted key K, there would be no need for the SM-DP to decrypt and then re-encrypt it. Instead, to maintain end-to-end key protection and minimize exposure of sensitive key material, the SM-DP would simply retain the encryption

on key K and include that encrypted key within the profile data it delivered to the eUICC, securely transported under the SM-DP Credential/session key.

179. A POSITA would also have understood that the symmetric key-wrapping key (DEK) used to encrypt key K must ultimately be available to the eUICC in order for the eUICC to decrypt the ciphertext and recover key K. A POSITA would have recognized that the MNO could transmit the DEK either directly to the eUICC, or via the SM-DP, as part of the profile-provisioning process. A POSITA would have understood that having the SM-DP send the DEK to the eUICC would be appropriate in situations where the MNO has delegated control over the timing of profile installation to the SM-DP, because relying on a single entity (the SM-DP) to provision all profile-related data is operationally more efficient than requiring the eUICC to receive data from two separate entities (the SM-DP and the MNO). In such implementations, the SM-DP would store the DEK together with the corresponding profile (including key K in ciphertext form) so that both could be securely provisioned to the eUICC during installation.

(2) the subscription manager's storage of the ciphertext and the symmetric key occurs before step a)

180. A POSITA would further have recognized that the SM-DP would logically prepare and store the profile (together with the symmetric key (DEK)

needed to decrypt the ciphertext), before the provisioning process is initiated. This sequencing would allow the SM-DP to prepare and stage the complete profile package in advance, enabling the provisioning session with the eUICC to proceed efficiently once initiated.

f. Claim 8: The method of claim 1, wherein after the step (b), the embedded universal integrated circuit card is authenticated by the subscription manager system using at least the identity of the embedded universal integrated circuit card included in the first message.

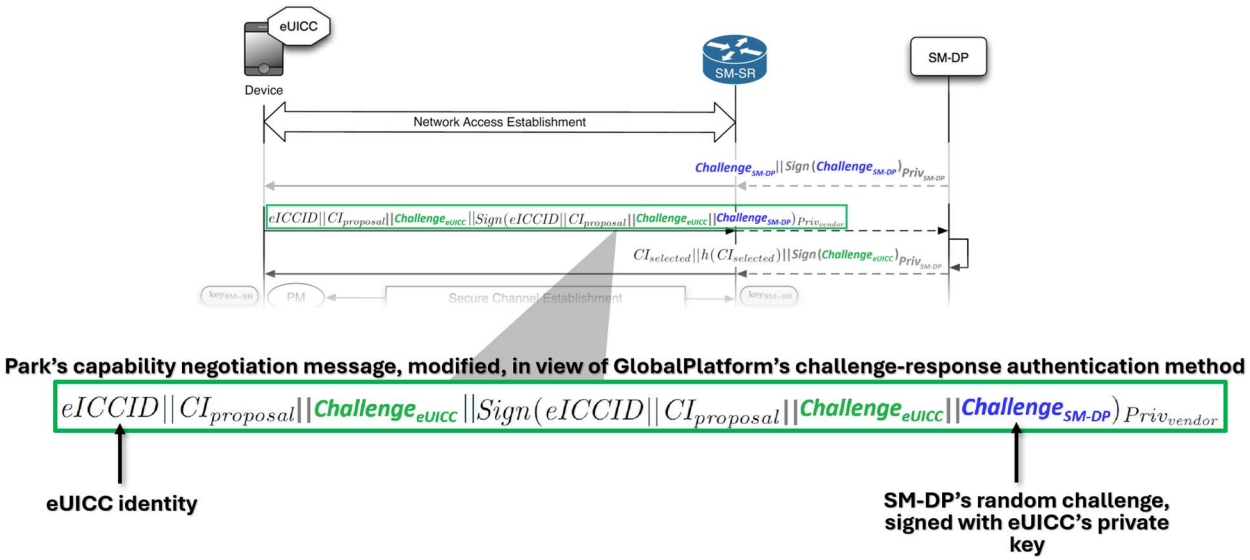
Claim 9: The method of claim 1, wherein after the step (b), the embedded universal integrated circuit card is authenticated by the subscription manager system using at least the digital signature included in the first message.

181. Park-GlobalPlatform-AbiChar teaches Claims 8-9.

182. Park teaches that, in response to receiving the capability negotiation message from the eUICC, the SM "acquires the eUICC Vendors certificate from the eUICC Certification Center," and, using that certificate, the "SM-DP verifies the signature of the eUICC ID and the capability information." Ex-1005, 300. A POSITA would have understood that the SM would use the eUICC ID included within the first message to identify the corresponding eUICC Vendors certificate.

183. As explained in Section XII.A.1 (motivation to combine), in the combined Park-GlobalPlatform-AbiChar system, Park's capability negotiation

process is modified to include GlobalPlatform's challenge-response authentication method, as illustrated below:



184. In the combined system, the capability negotiation message sent from the eUICC (claimed "first message") includes the eUICC's response to a random challenge ($Challenge_{SM\text{-}DP}$) issued by the SM-DP—namely, the SM-DP's challenge signed/encrypted with the eUICC's private key ($Sign(\ldots Challenge_{SM\text{-}DP})Priv_{vendor}$) (claimed "digital signature"). Ex-1006, 246-47, 266-68, 271. To authenticate the eUICC, the SM-DP uses the eUICC's public key—retrieved from the eUICC Vendors certificate—to decrypt the signed challenge. If the decrypted challenge matches the challenge issued to the eUICC, this verifies that the eUICC is the authentic holder of the eUICC private key, thereby authenticating the eUICC to the SM-DP.

185. Thus, in the combined system, the SM-DP would use both the eUICC ID (Claim 8) and the eUICC's digital signature (Claim 9) from the capability negotiation message received from the eUICC to authenticate the eUICC.

g. Claim 10: The method of claim 1, wherein the decrypted ciphertext is stored in nonvolatile memory of the embedded universal integrated circuit card.

186. Park-GlobalPlatform-AbiChar teaches Claim 10.

187. As explained for Element 1[j], Park teaches that, following successful decryption of the profile blocks—which would include decryption of the ciphertext containing key K encrypted with the symmetric key—"the profile blocks are provisioned inside the eUICC." Ex-1005, 300. A POSITA would have understood that this provisioning process would involve storing the decrypted profile data, including key K, within the eUICC's non-volatile (i.e., persistent) memory so that it remains available for use in accessing the associated mobile network after power loss or device reset.

h. Claim 11: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module and wherein the module is a machine-to-machine module.

188. Park-GlobalPlatform-AbiChar teaches Claim 11.

189. Park teaches that an initial use for eUICCs was "for the M2M (Machine-to-Machine) device." Ex-1005, 297.

i. Claim 12: The method of claim 11, wherein the module comprises a module of a type selected from the group consisting of: vending machine, automobile, alarm system, tracking device, monitoring device, security system and remote control device.

190. Park-GlobalPlatform-AbiChar teaches Claim 12.

191. Park teaches that eUICCs have been "adopted into the small device such as smart meter, etc." Ex-1005, 297. A POSITA would have understood that a smart meter is a type of monitoring device.

j. Claim 13: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module and wherein the module is a wireless module.

Claim 14: The method of claim 13, wherein the module is configured to communicate using one or more wireless technologies of types selected from the group consisting of: WiFi, WiMax, General Packet Radio Services, Enhanced Data rates for GSM Evolution and 3rd Generation Partnership Project technology.

192. Park-GlobalPlatform-AbiChar teaches Claims 13-14.

193. Park teaches that the eUICC communicates with the SM using a "Device Module (DM)," as illustrated in annotated Figure 5, below. Ex-1005, 299.

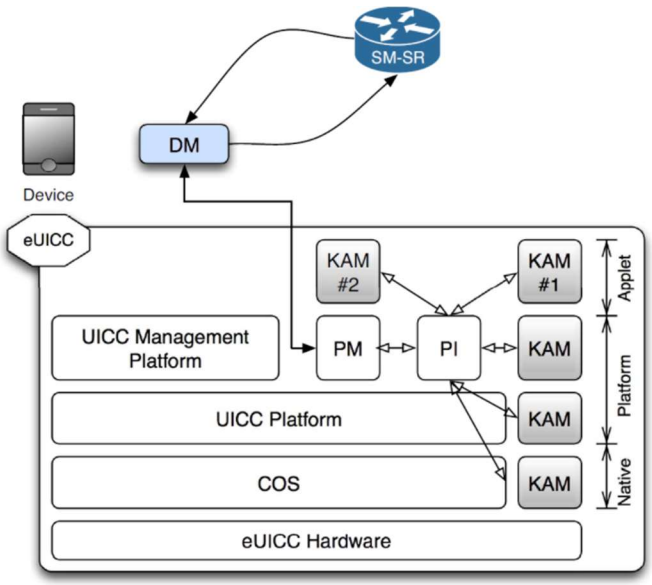


Fig. 5. Internal Architecture of eUICC

194. Park further teaches that this communication may occur over a wireless network, such as "WiFi," or a "mobile network" accessible to the eUICC through "the provisioning profile." Ex-1005, 300.

k. Claim 15: The method of claim 13, wherein the module comprises a module of a type selected from the group consisting of the following: wireless handset, cellular phone, smartphone, laptop computer, tracking device, computer with a radio, circuit board with a radio, hand-held device, netbook, portable computer, multiprocessor system, microprocessor based consumer electronic, programmable consumer electronic, network personal computer, minicomputer, mainframe computer and smartcard.

195. Park-GlobalPlatform-AbiChar teaches Claim 15.

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

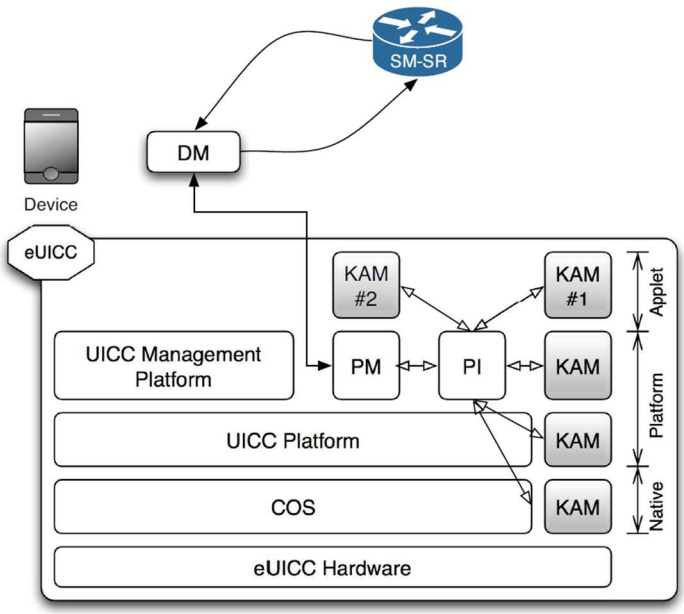


Fig. 5. Internal Architecture of eUICC

l. Claim 16: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module, wherein the module is a wired module.

Claim 17: The method of claim 16, wherein the module is configured to communicate using one or more wired connections of types selected from the group consisting of: Ethernet, fiber optic connection and Universal Serial Bus connection.

197. Park-GlobalPlatform-AbiChar teaches Claims 16-17.

198. Park teaches that the eUICC may communicate with the SM-SR using "USB[], without a Provisioning Profile." Ex-1005, 300. A POSITA would have recognized that a USB interface is a wired interface.

m. Claim 19: The method of claim 1, wherein the deriving step (c) is performed prior to the generating step (a).

Claim 20: The method of claim 1, wherein the generating step (a) is performed before the deriving step (c).

199. Park-GlobalPlatform-AbiChar teaches Claims 19-20.

200. A POSITA would have understood that the timing of when the eUICC generates its ephemeral key pair for use in the key exchange—whether before or after sending the initial capability negotiation message—would have been a simple matter of design choice.

201. A POSITA would have recognized that a designer could implement the ephemeral key generation either (1) close in time to when those keys are needed by the key exchange algorithm, or (2) earlier in the process (e.g., before sending the capability negotiation message), and that the choice between the two would involve a tradeoff between security and performance. For instance, generating the keys immediately before the key exchange increases security by limiting the private key's lifetime and thereby reducing its exposure, but can introduce delay into the provisioning process. On the other hand, generating the keys earlier avoids such

delays, but extends the private key's lifetime and therefore increases security risk. A POSITA would have appreciated that system designers could flexibly adopt either approach, depending on whether minimizing delay or maximizing key protection was the higher operational priority.

n. Claim 21: The method of claim 1, wherein the embedded integrated circuit card is operably connected to a module and a user accesses a user interface of the module to enter authentication information before the module receives the symmetric key.

202. Park-GlobalPlatform-AbiChar teaches Claim 21.

203. 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, 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.

204. 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.

o. Claim 22: The method of claim 1, wherein the subscription manager receives the ciphertext from a mobile network operator before the network application sends the first message.

205. Park-GlobalPlatform-AbiChar teaches Claim 22 for the same reasons as explained for Claim 7.

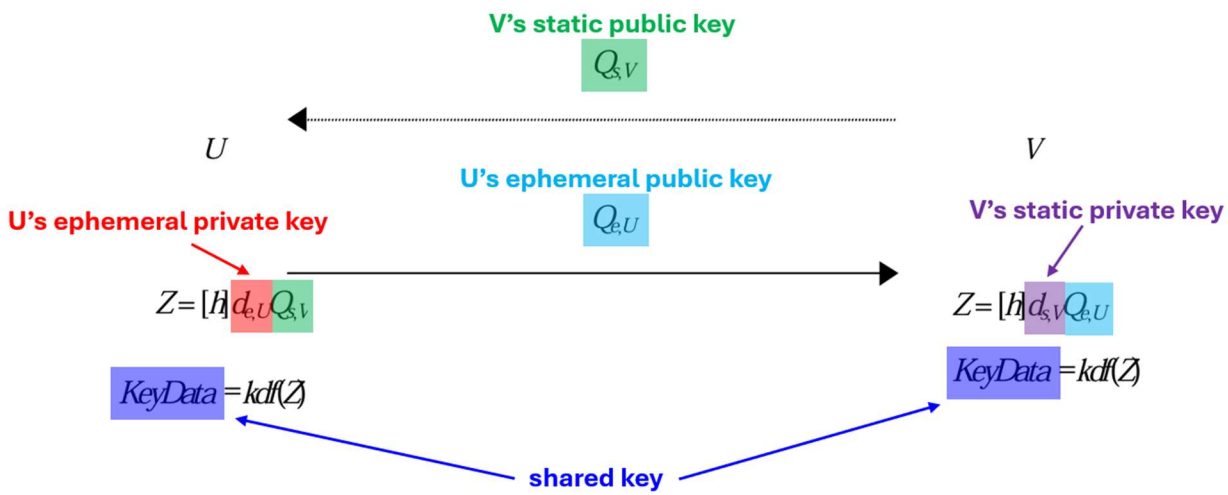
B. Ground 2: Claims 1-17 and 19-22 are obvious over Park in view of GlobalPlatform and X9.63-Overview.

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.

206. A POSITA would have been motivated to combine Park's secure profile provisioning architecture (SPA) with GlobalPlatform's procedures for mutual authentication and 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, 299.

207. As X9.63-Overview illustrates, the parties exchange public keys, Q_{sV} and Q_{eU}, and then use the corresponding private keys a temporary **private key $d_{e,U}$** and **corresponding public key $Q_{e,U}$** derived from a random number:

ANSI X9.63 - 1-Pass DH



Ex-1008, 12.

208. After deriving a shared secret Z, each party can derive the same symmetric key, KeyData, corresponding to the profile key, as described for 1(e). 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.

2. Independent Claim 1

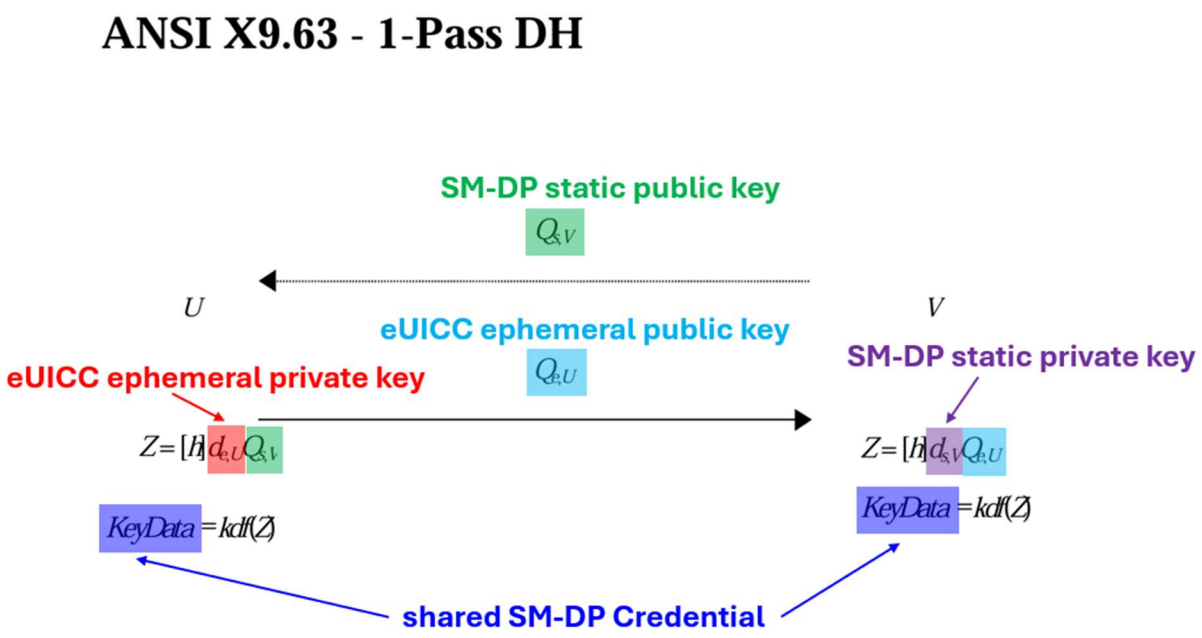
a. Elements 1[pre]-1[b] and 1[f]-1[j]

209. Park in view of GlobalPlatform and X9.63-Overview ("Park-GlobalPlatform-X9.63-Overview") teaches Elements 1[pre] through 1[b], and 1[f] through 1[j], for the same reasons explained for Ground 1.

b. Element 1[c]: (c) deriving a second eUICC private key and a corresponding second eUICC public key using a first random number generator and a first set of cryptographic algorithms;

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

211. As explained in Section XII.B.1 (motivation to combine), a POSITA would have been motivated to use X9.63-Overview's 1-pass Diffie-Hellman key exchange, illustrated below, to implement Park's "asymmetric algorithm-based key agreement protocol," as illustrated below:



Ex-1007, 12.

212. As illustrated, in the combined Park-GlobalPlatform-X9.63-Overview system, the eUICC generates an ephemeral private key $d_{e,U}$ (claimed "second eUICC private key"). *Id.* X9.63-Overview teaches that ephemeral private key $d_{e,U}$ is "selected randomly." *Id.*, 7. A POSITA would have understood that randomly selecting a number involves using a random number generator. The eUICC then calculates a corresponding ephemeral public key $Q_{e,U}$ (claimed "second eUICC public key"). Ex-1007, 3. X9.63-Overview "[s]pecifies key agreement and key transport schemes using elliptic curve cryptography." Ex-1008, 3. Accordingly, a POSITA would have understood that calculating public key $Q_{e,U}$ (claimed "second

eUICC public key") would involve applying the appropriate elliptic-curve cryptographic algorithm to the randomly selected private key $d_{e,U}$.

c. Element 1[d]: (d) storing a subscription manager public key which corresponds to a subscription manager private key;

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

214. As explained in Section XII.B.1 (motivation to combine), in the combined system, X9.63-Overview's 1-pass Diffie-Hellman protocol employs SM-DP's static public-private key pair—i.e., static private key $d_{s,V}$ (claimed "subscription manager private key"), and static public key $Q_{s,V}$ (claimed "subscription manager public key"). Ex-1007, 12. This scheme assumes that the eUICC has been pre-provisioned with (i.e., stores) the **SM-DP's static public key**. Ex-1008, 12. Indeed, in the combined system, the eUICC would use the same **SM-DP static public key** to calculate the shared SM-DP Credential/session key as it would have used prior to engaging in the key exchange, to authenticate the SM-DP (by verifying the SM-DP's private key signature on the challenge it issued to the SM-DP). *See* Section XII.A.2.e.

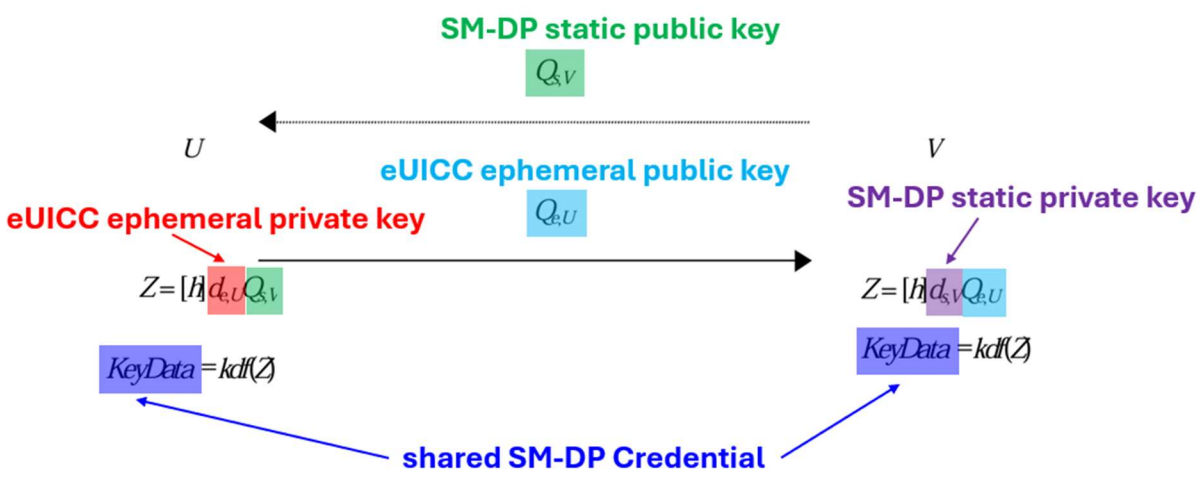
d. Element 1[e][1]: (e) deriving a profile key using a key exchange algorithm based on at least:

(i) the second eUICC private key, and

(ii) the stored subscription manager public key,

215. Park-GlobalPlatform-X9.63-Overview teaches Element 1[e][1].

216. In the combined system, each party uses its private key and the other party's public key to generate a shared SM-DP Credential/session key (claimed "profile key") as illustrated in the diagram below:



Ex-1008, 12.

217. Thus, the eUICC calculates the **shared SM-DP Credential/session key** (claimed "profile key") using its private key $d_{e,U}$ (claimed "second eUICC

private key"), together with the stored SM-DP public key $Q_{s,V}$ (claimed "subscription manager public key). *Id*.

e. Element 1[e][2]: wherein the profile key can also be derived at the subscription manager system based at least on: the second eUICC public key, and the subscription manager private key;

218. Similarly, the SM-DP calculates the **shared SM-DP Credential/session key** (claimed "profile key") using its private key $d_{s,V}$ (claimed "subscription manager private key"), together with the eUICC public key $Q_{e,U}$ (claimed "second eUICC public key) that it received from the eUICC. Ex-1008, 12.

219. Thus, Park-GlobalPlatform-X9.63-Overview teaches Element 1[e][2].

220. Accordingly, Park-GlobalPlatform-X9.63-Overview renders obvious Claim 1.

3. Dependent Claims 2-17, 19-22

a. Claims 2-3, 6-17, 19-22

221. Park-GlobalPlatform-X9.63-Overview teaches Claims 2-3, 6-15, 17-20 for the same reasons explained for Ground 1.

b. Claim 4: The method of claim 1, wherein the key exchange algorithm is a Diffie-Hellman key exchange.

Claim 5: The method of claim 1, wherein the key exchange algorithm is an Elliptic Curve Diffie-Hellman key exchange.

222. Park-GlobalPlatform-X9.63-Overview teaches Claims 4-5.

223. X9.63-Overview teaches a "1-Pass" Diffie-Hellman key exchange (Ex-1008, 12) that is included in ANSI X9.63's set of "key agreement and key transport schemes using elliptic curve cryptography" (Ex-1008, 3). Thus, X9.63-Overview teaches a Diffie-Hellman key exchange that uses "elliptic curve cryptography"—i.e., an Elliptic Curve Diffie-Hellman key exchange. Ex-1008, 3, 12.

C. Grounds 3-4: Claim 18 is obvious over Park in view of GlobalPlatform, AbiChar, and Weiss, and Park in view of GlobalPlatform, X9.63-Overview, and Weiss

1. A POSITA would have been motivated to add Weiss's teachings to the Park-GlobalPlatform-AbiChar and Park-GlobalPlatform-X9.63-Overview systems and would have had a reasonable expectation of success.

224. Weiss teaches that "secure elements, such as SIMs, eUICCs, UICCs" (Ex-1009, ¶3) may be included in a variety of devices, "such as a tablet or notebook computer, a TV system, a set top box, a vending machine, a vehicle, a surveillance camera, a sensor device and the like" (*id.*, ¶31). A POSITA seeking to broaden the applicability of Park's eUICC provisioning architecture beyond the specific implementations identified in Park would have looked to references such as Weiss, which expressly teaches that secure elements can be embedded in numerous types of connected or networked devices. Incorporating Park's eUICC into Weiss's disclosed devices would have amounted to a straightforward substitution of one known secure element implementation for another, yielding predictable results.

2. Claim 18: The method of claim 16, wherein the module comprises a module of a type selected from the group consisting of: computer, security camera, security monitoring device and networked controller.

225. Park-GlobalPlatform-AbiChar-Weiss and Park-GlobalPlatform-X9.63-Overview-Weiss teach Claim 18.

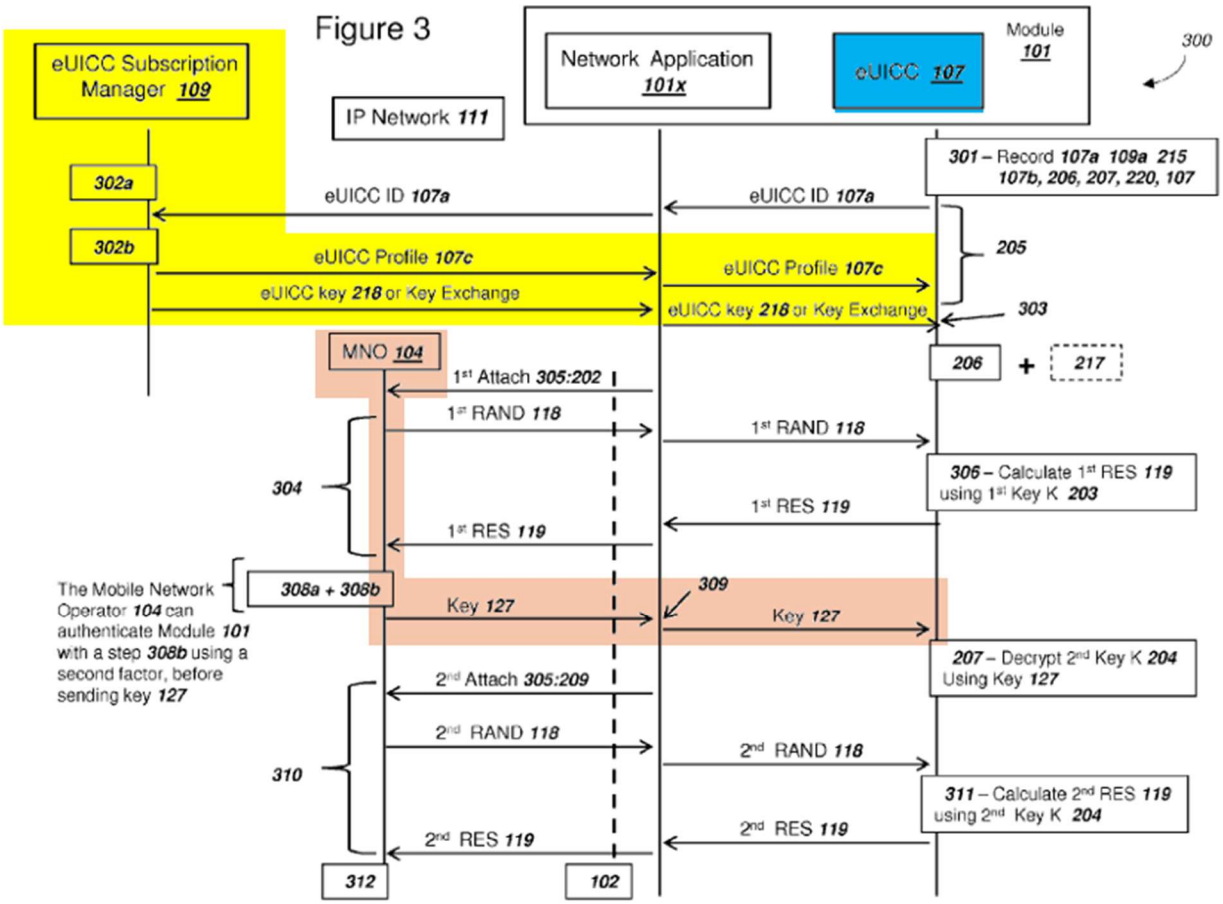
226. Weiss teaches that an eUICC may be included in "a tablet or a notebook computer." Ex-1009, ¶31.

D. Ground 5: Claims 1-22 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.

227. Nix175 discloses the claims of the '094 Patent with the exception of claim element 1(g) "receiving the symmetric key," to the extent PO alleges that receiving the symmetric key includes receiving it from the subscription manager (SM). As discussed above in Sections **Error! Reference source not found.** and **Error! Reference source not found.**, the '094 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:



228. Above, the subscription manager (yellow) provides the encrypted eUICC profile 107c to the eUICC (blue). But the eUICC cannot decode secret key K ("2nd Key K 204) until it receives symmetric key 127 (red) from the MNO 104. Ex-1016, Fig. 3.

229. 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.

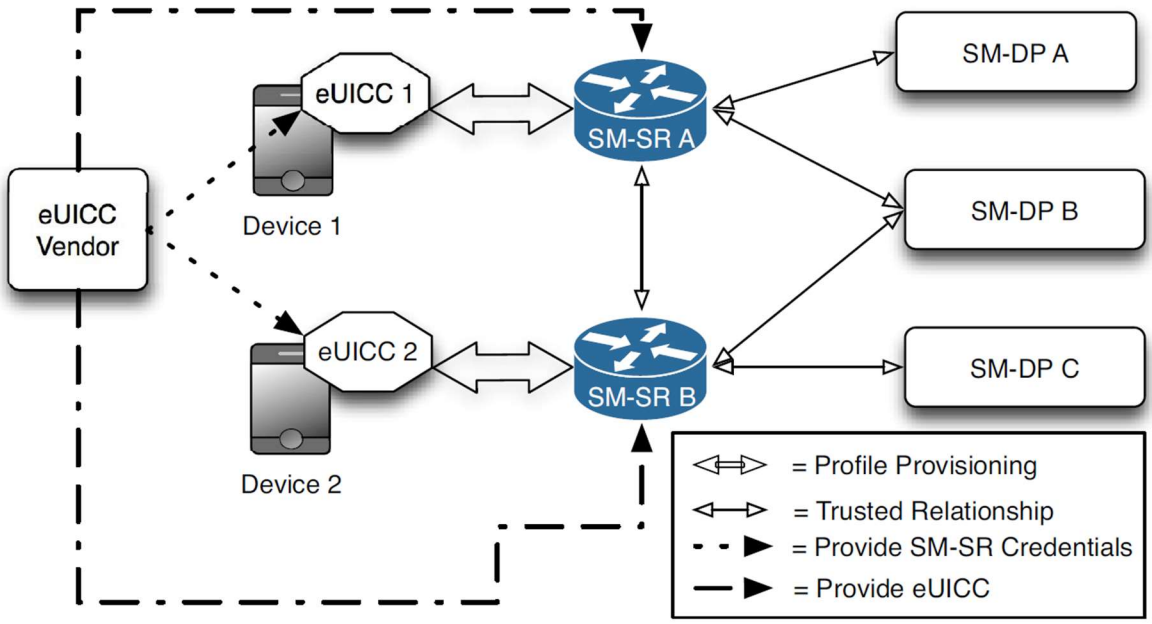


Fig. 1. eUICC Profile Ecosystem

230. By splitting the SM, Park allows for SM-SR, which securely routes the encrypted profile to the eUICC, to exist in a different trust zone[4] and potentially be 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 secret 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.

231. In this combined system, a POSITA also would have recognized advantages to continuing to encrypt secret key K with a symmetric key that is separate from the key used to encrypt the overall profile.

232. ***First***, 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 would not be able to pre-encrypt the profile with the profile key. At the same time, a POSITA would have understood

[4] "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.

that sensitive data, such as secret key K, should not be stored in Park's 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 Park's SM-DP could store and handle key K securely, before establishing the profile key with the eUICC.

233. ***Second***, a POSITA would 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").

234. 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.

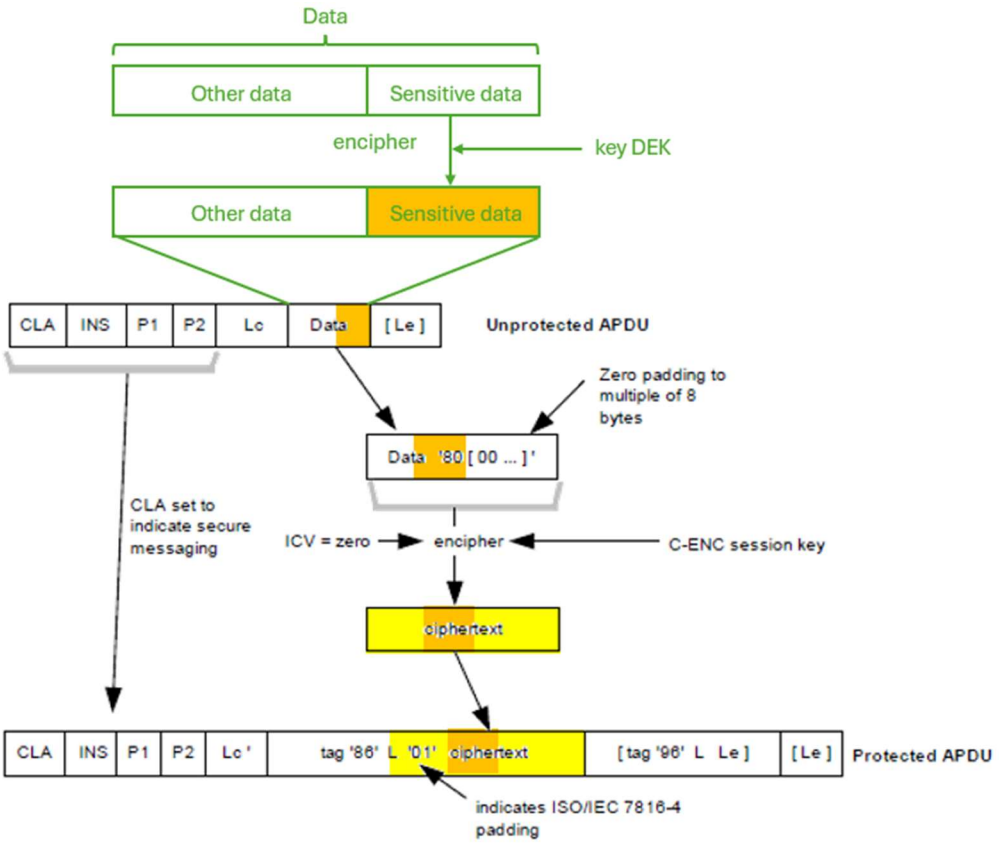


Figure F-11: Secure Messaging: Command message protected for confidentiality

235. 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.

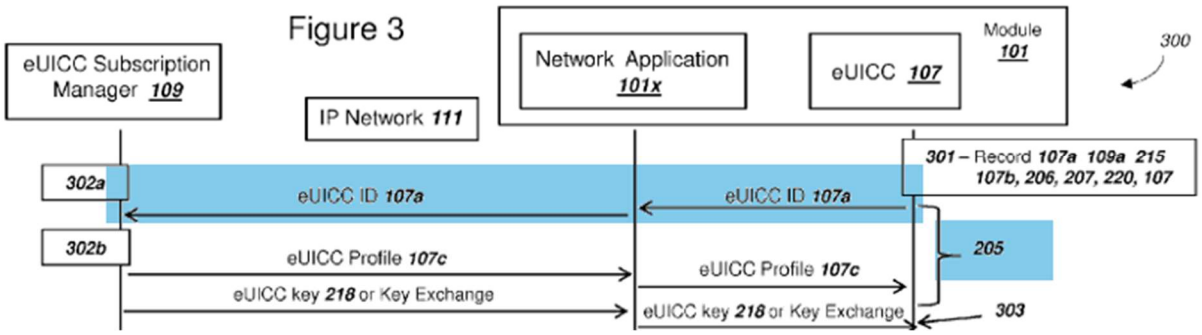
2. Independent Claim 1

a. Element 1[pre]: A method comprising:

236. Nix175 teaches "[a] method for authentication, the method performed by a hardware module using an embedded Universal Integrated Circuit Card (eUICC)." Ex-1016, Cl. 1.

b. Element 1[a][1]: (a) generating a first message comprising (1) an identity of an embedded universal integrated circuit card (eUICC); (2) a nonce; and (3) a first digital signature, generated using a first eUICC private key, wherein the first eUICC private key corresponds to a first eUICC public key;

237. As shown in Figure 3, excerpted below, Nix175's eUICC generates a first message at step 205 that is sent to the Subscription Manager 109:



Ex-1016, Fig. 3. This message includes (1) the eUICC identity 107a, (2) a nonce, and (3) a digital signature, where the digital signature is generated using the eUICC private key:

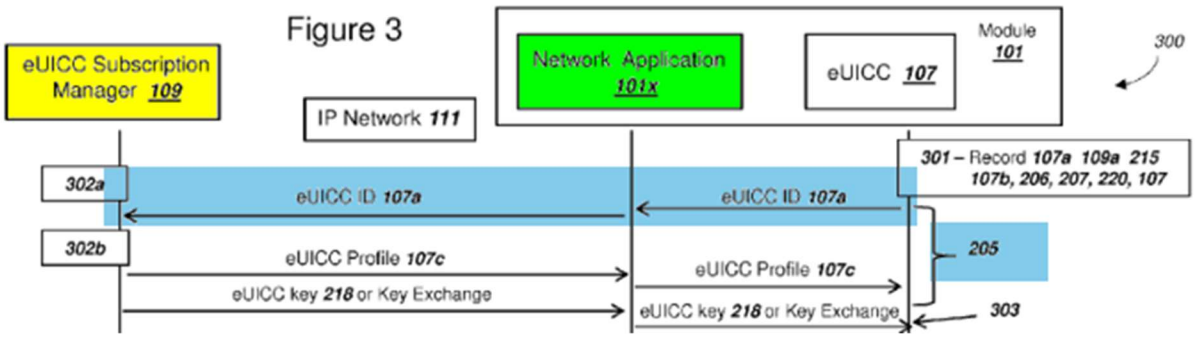
> In an exemplary embodiment, a message with ***the eUICC identity 107a*** in a step 205 could include ***a digital signature***, where the digital

> signature is processed using (i) ***the eUICC private key 215*** recorded in a step 301 and (ii) a digital signature algorithm 221. The message with the eUICC identity 107a and a digital signature in a step 205 could ***preferably include a random number or string, including a nonce or a "number used once"*** such as, but not limited to, an exemplary security token 212 in order to prevent replay attacks.

Id. 48:65-49:6. Nix175 explains that "eUICC private key 215 [] can be associated with an eUICC public key 214," providing the specific example where "eUICC private key 215 and eUICC public key 214 [] comprise a public key infrastructure (PKI) key pair." *Id.* 41:40-44. Nix175 teaches that the subscription manager stores a copy of the eUICC's public key (*id.*, 47:52-55), and, in response to receiving this message, uses it to authenticate the eUICC, by verifying the eUICC's digital signature. Ex-1016, 43:6-19, 49:15-19.

c. Element 1[b]: (b) sending the first message via a network application to a subscription manager system;

238. As shown in Figure 3, Nix175 shows that the message in step 205 is sent via a network application 101x (green) to a subscription manager 109 (yellow):



Ex-1016, Fig. 3.

d. Element 1[c]: (c) deriving a second eUICC private key and a corresponding second eUICC public key using a first random number generator and a first set of cryptographic algorithms;

239. Nix175 discloses that the eUICC private/public key pair may be derived using a random number generator and cryptographic algorithms: "Several possibilities exist for the source of an eUICC private key 215 and eUICC public key 214. The ***eUICC private key*** 215 and ***eUICC public key*** 214 be generated using standard software tools." Ex-1016, 42:11-14. "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." *Id.* 42:25-30.

240. Nix fails to disclose a second private/public eUICC key pair (different from eUICC private key 215 used for the digital signature in element 1[a]). But a POSITA would have recognized security advantages to deriving a second eUICC private/public key pair for use in the ECDH key-exchange (as described below for Element 1[e][1]), separate from the eUICC private/public key pair used for authentication (as described for Element 1[c][1]). In particular, it would have been well known to a POSITA that a key should only be used for a single purpose, to limit

its exposure risk. *See, e.g.*, Ex-1023, NIST SP 800-57, 45 ("In general, a single key **should** be used for only one purpose.").

e. Element 1[d]: (d) storing a subscription manager public key which corresponds to a subscription manager private key;

241. Nix175 teaches "As illustrated in FIG. 2d, the eUICC subscription manager 109 may also be associated with an eUICC ***subscription manager public key 220*** and an ***eUICC Subscription manager private key 222***, and the two keys can comprise a ***PKI key pair*** for the eUICC subscription manager 109." Ex-1016, 43:20-24. Nix175 further teaches "module 101 can record an eUICC identity 107a, an address of an eUICC subscription manager 109, and an eUICC private key 215." *Id.* 47:15-17. And "[o]ther data could be included in a step 301 as well, such as, but not limited to, … an eUICC public key 214, [and] an ***eUICC subscription manager public key 220***." *Id.* 47:22-25. As explained for Element 1[c][5][i], below, the subscription manager public key is used in a key exchange, to derive the profile key. In the combined system, in which Park's SM-DP is trusted with encrypting the profile, Nix175's eUICC subscription manager public key would correspond to the SM-DP public key.

f. Element 1[e][1]: (e) deriving a profile key using a key exchange algorithm based on at least:

(i) the second eUICC private key, and

(ii) the stored subscription manager public key,

242. 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[5]. 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).

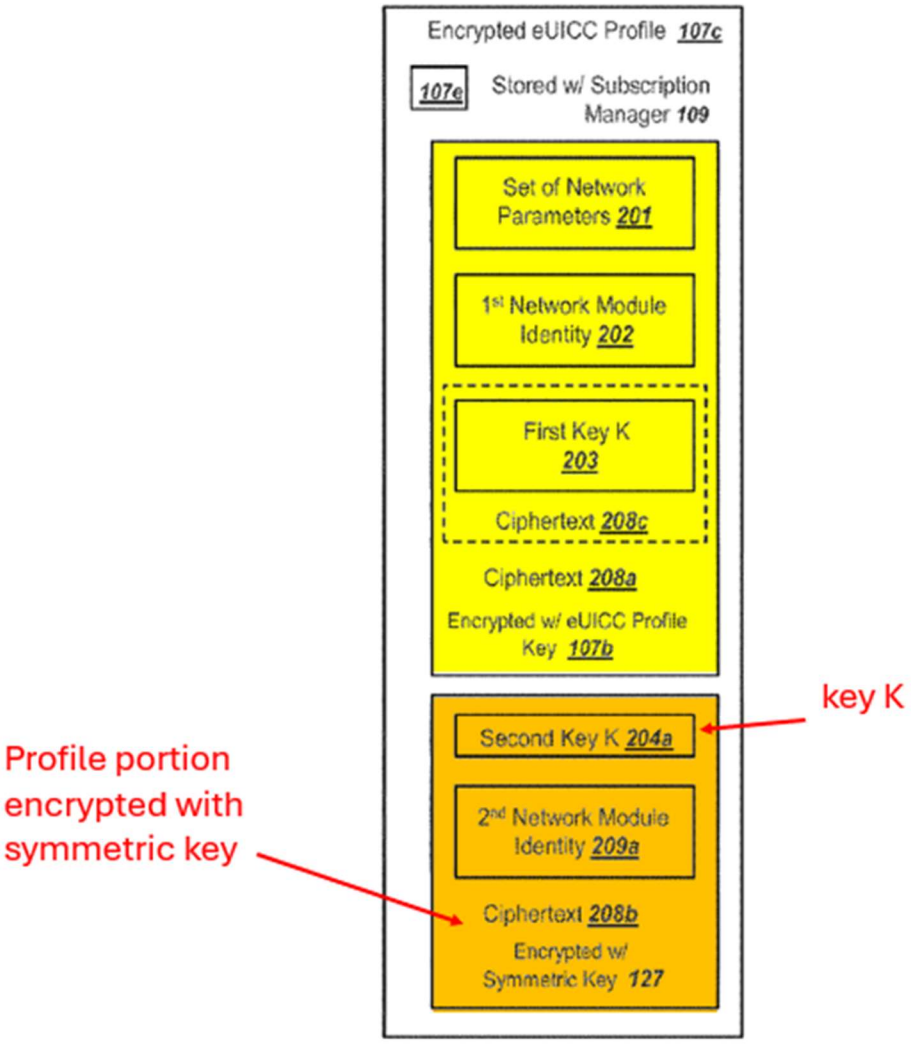
[5] For example, Nix175 cites X-9.63 as describing an exemplary key exchange. Ex-1016, 52:22-23; *see* Ex-1007, 12 (describing DH key exchange).

g. Element 1[e][2]: wherein the profile key can also be derived at the subscription manager system based at least on: the second eUICC public key, and the subscription manager private key;

243. As discussed above, the same profile key can also be calculated by B (subscription manager) using B's private key n_b (subscription manager private key) and A's public key Y_a (eUICC public key), such that $K = n_a Y_b = n_b Y_a = n_a n_b B$. Ex-1007, 3. Thus, the same profile key can also be derived by the subscription manager using its private key and the eUICC public key.

h. Element 1[f]: (f) receiving from the subscription manager system, an encrypted profile comprising a ciphertext including a key K encrypted with a symmetric key;

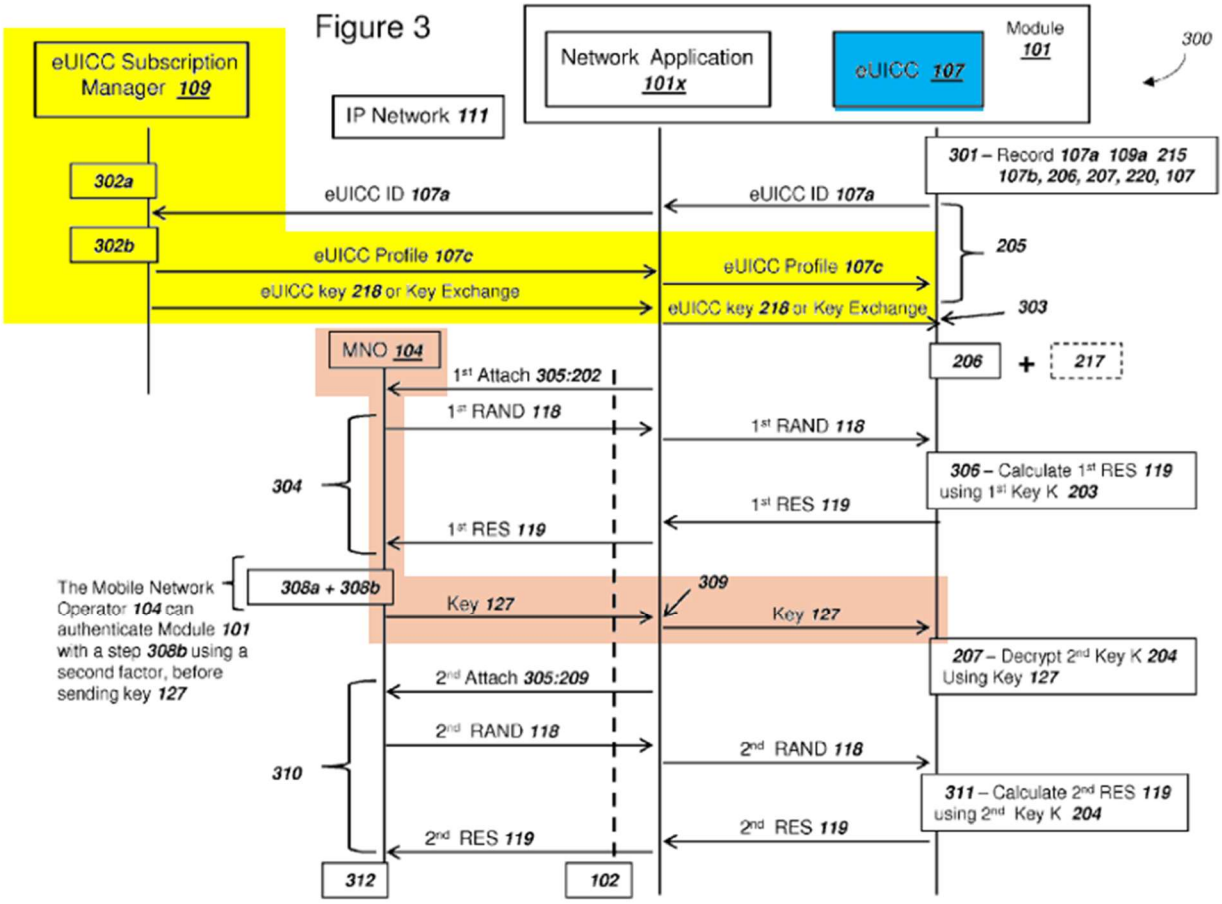
244. Nix175 teaches "receiving, by the hardware module, a profile from the eUICC subscription manager … wherein the hardware module uses the symmetric key to decrypt the second portion of the profile, and wherein the second portion includes a second key K2." Ex-1016, Cl. 1. The portion of the profile encrypted with the symmetric key and comprising ciphertext 208b including "second key K 204a" (the claimed "key K"), is shown in orange in the annotated excerpt of Figure 2a, below.



Ex-1016, Fig. 2a, 40:9-13.

i. Element 1[g]: (g) receiving the symmetric key;

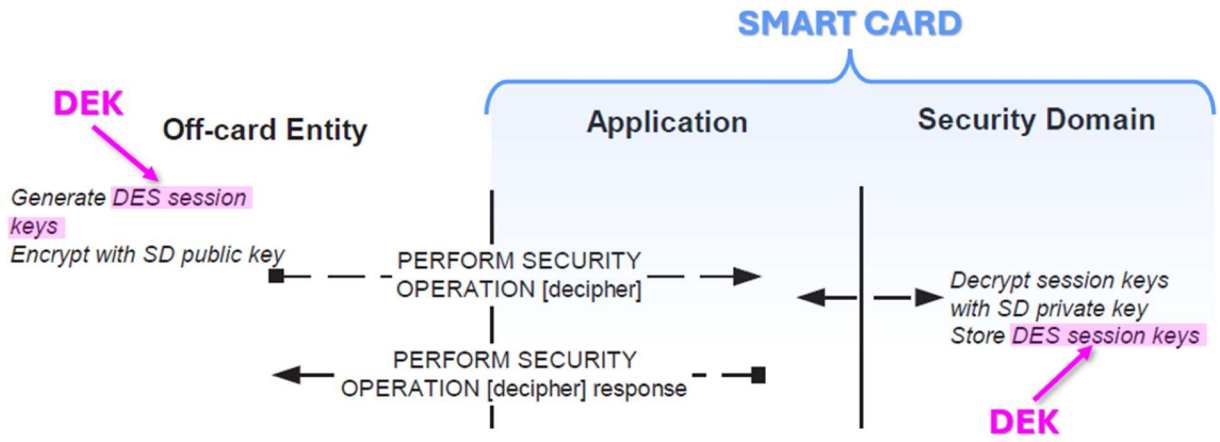
245. As discussed above, Nix175 teaches that the ***MNO*** sends the symmetric key 127 used to encrypt the key K in the profile, as shown below in red.



246. 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.D.1.

247. 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:

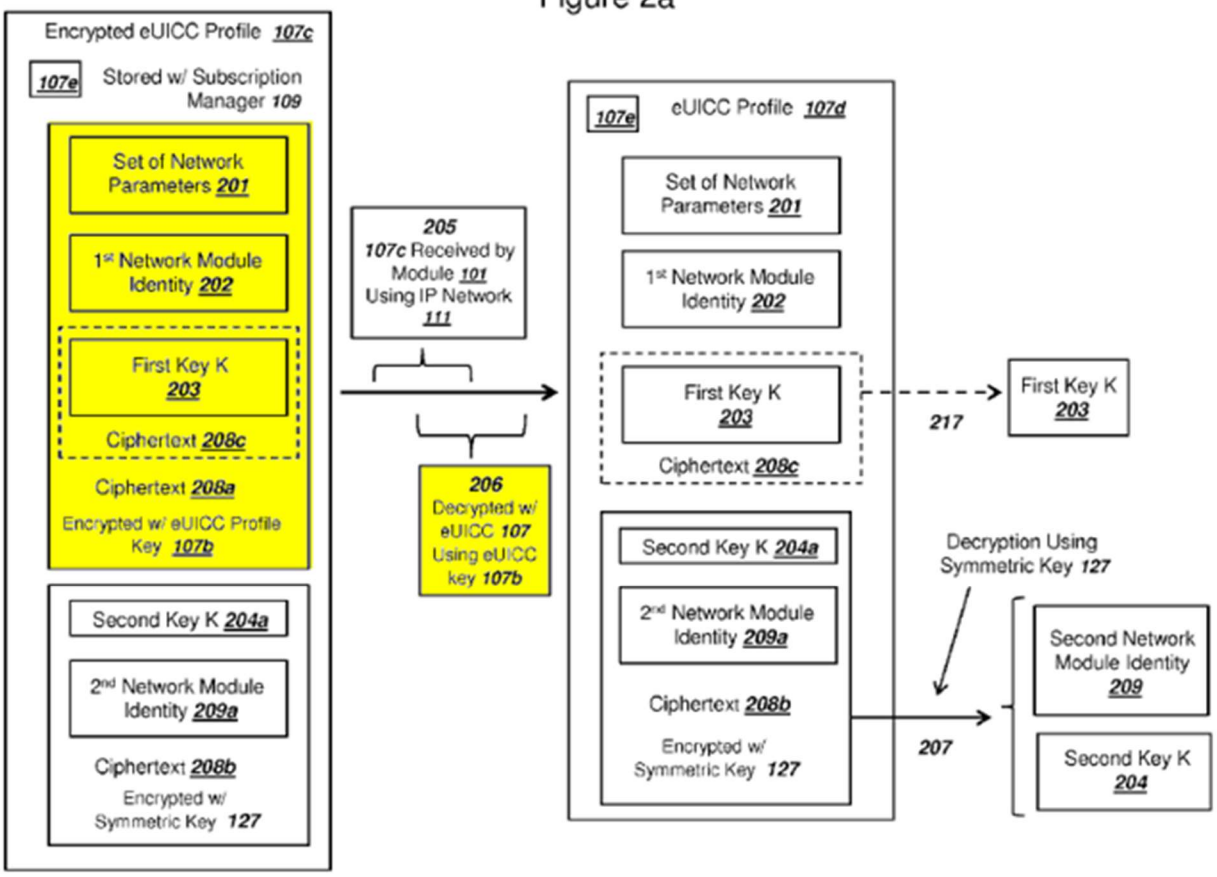


248. A POSITA would have recognized that in the combined Nix175-Park-GlobalPlatform system, Park's SM-DP would send the DEK (claimed "symmetric key") to the eUICC. Further, the SM-DP is part of the network trust zone separate from the SM-SR. Thus, a POSITA would have recognized that in the Nix175-Park-GlobalPlatform 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).

j. Element 1[h]: (h) decrypting, by the embedded universal integrated circuit card, at least a portion of the encrypted profile using the profile key;

249. 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.

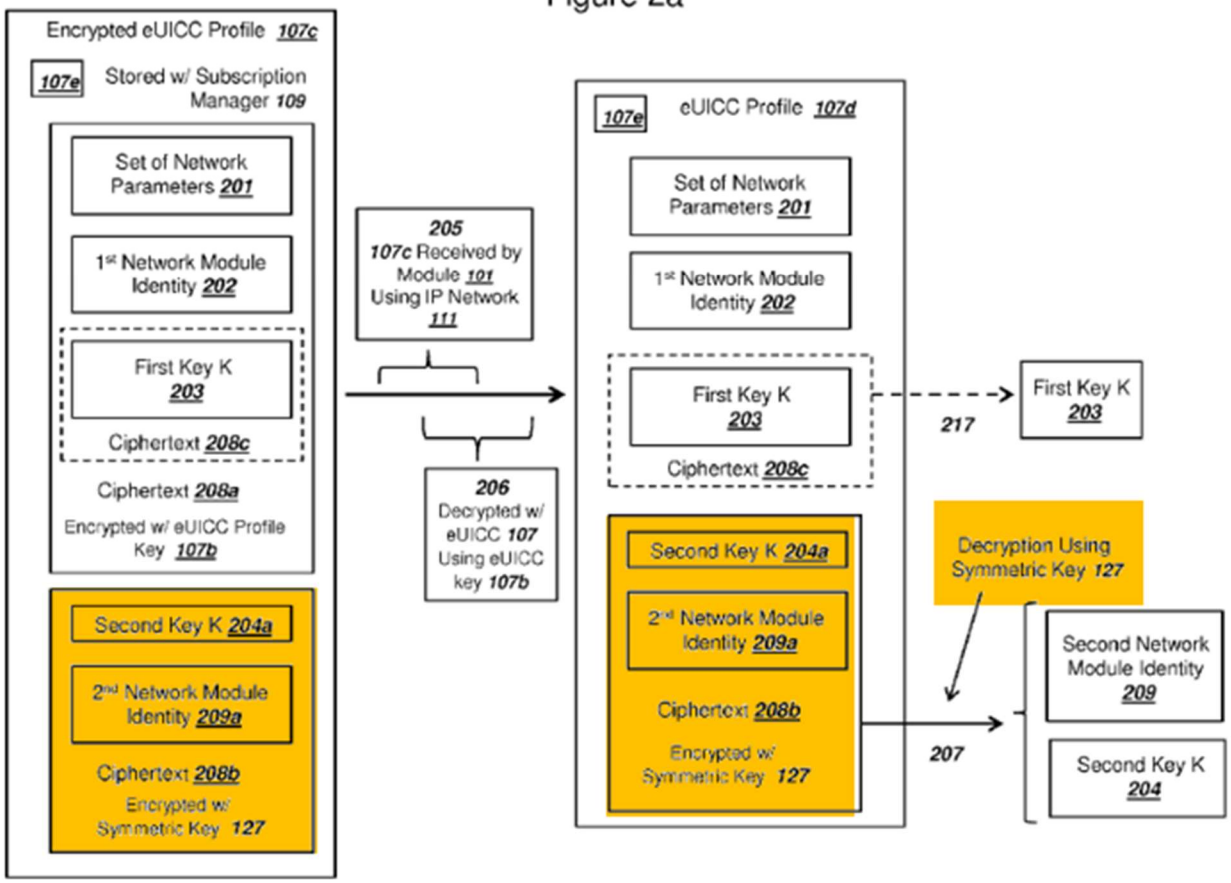
Figure 2a



k. Element 1[i]: (i) decrypting, by the embedded universal integrated circuit card, at least a portion of the ciphertext using the symmetric key; and

250. 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 encrypted by symmetric key 127 and then being decrypted by symmetric key 127 at step 207. Ex-1016, Fig. 2a.

Figure 2a



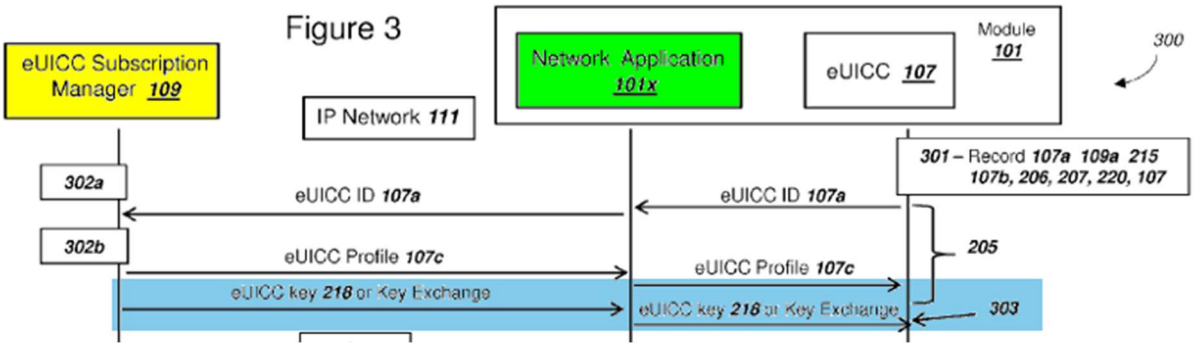
l. Element 1[j]: (j) storing at least the key K in the embedded universal integrated circuit card for use in future communications.

251. Nix175 teaches that the "second key K 204 can also be used with standards-based authentication with a wireless network 102, where the second key K 204 in FIG. 2a is also depicted and described as operating as a standards-based shared secret key K in FIG. 3 below at a step 311." Ex-1016, 32:31-35. Further, Nix175 teaches "wherein the UICC records the second key K2 in a protected nonvolatile memory after the eUICC decrypts the second key K2." *Id.* Cl. 8. Thus, a POSITA would have understood the claimed "key K" would have been stored in the eUICC for use in future communications, according to the wireless communication standards.

3. Dependent Claims 2-22

a. Claim 2: The method of claim 1, wherein the subscription manager public key is downloaded via the network application.

252. Nix175 teaches that the "ECDH key exchange in a step 303," illustrated below, includes "eUICC subscription manager public key 220." Ex-1016, 51:65-52:2.



253. Nix175 further teaches that this subscription manager public key is "recorded with an eUICC 107 in a step 301" (*id.*, 49:49-55), which may occur at the time of manufacture, or, alternatively, "the data could be downloaded or accessed separately via an IP network 111." *Id.*, 47:18-22. A POSITA would have understood that this download would occur from the subscription manager (yellow) via network application 101x (green). Ex-1016, Fig. 3, 46:7-29.

b. Claim 3: The method of claim 1, wherein the step of storing the subscription manager public key occurs at the time of manufacture of the embedded universal integrated circuit card.

254. Nix175-Park-GlobalPlatform discloses claim 3. As explained for Claim 2, above, Nix175 teaches that the subscription manager public key (corresponding to Park's SM-DP public key in the combined system) is "recorded with an eUICC 107 in a step 301" (*id.*, 49:49-55), which may be performed by a "manufacturer" at the time of manufacture. *Id.*, 47:18-22. Similarly, Park teaches that the "eUICC Vendor develops eUICCs. During this, the SM-SR credentials

($key_{SM\text{-}SR}$) and the capability information ($CI_{installed}$) of each eUICC are installed into each eUICC." Ex-1005, 300. A POSITA would have understood that, in the combined system, Park's eUICC Vendor would store both SM-SR credentials and the SM-SD credentials (in the form of an SM-SD public key).

c. Claim 4: The method of claim 1, wherein the key exchange algorithm is a Diffie-Hellman key exchange.

Claim 5: The method of claim 1, wherein the key exchange algorithm is an Elliptic Curve Diffie-Hellman key exchange.

255. Nix175 teaches that "An eUICC 107 within a module 101 could use a ECDH [Elliptic Curve Diffie Hellman] key exchange in a step 303 where 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; Fig. 3. Thus Nix175 teaches the key exchange algorithm is a Diffie-Hellman exchange and specifically, an elliptic curve Diffie-Hellman exchange.

d. Claim 6: The method of claim 1, wherein the encrypted profile is encrypted in a manner so that intermediate nodes on an IP network would not be able to read data within the profile in an unencrypted form.

256. Nix175 teaches "eUICC subscription manager 109 sends the profile 107c, where the network access credentials (including the first key K 203) are encrypted with an eUICC profile key 107b, and in this manner intermediate nodes

on the IP network 111 would not feasibly be able to read the data within the profile 107c." Ex-1016, 51:24-28.

e. Claim 7: The method of claim 1, wherein subscription manager system stores the ciphertext and the symmetric key before step (a).

257. Nix175-Park-GlobalPlatform teaches Claim 7.

258. Park's SM-DP prepares the profiles to be sent to the eUICC, and as explained in Section **Error! Reference source not found.** of Ground 1, GlobalPlatform taught to separately encrypt sensitive key data with DEK. Because Park's SM-DP can encrypt a portion of the profile with DEK (the claimed "symmetric key") without needing to know the eUICC destination, it can do so, and store DEK, before step (a), when the eUICC sends its eUICC ID. It would have been a simple matter of design choice of whether to pre-encrypt profiles with the symmetric key to speed the process of provisioning at the cost of increased storage requirements in the SM-DP. Moreover, Nix175 teaches "eUICC subscription manager 109 could assemble and cipher the profile 107d into a profile 107c at an earlier time, such as before receiving the eUICC identity 107a." Ex-1016, 50:57-59.

f. Claim 8: The method of claim 1, wherein after the step (b), the embedded universal integrated circuit card is authenticated by the subscription manager system using at least the identity of the embedded universal integrated circuit card included in the first message.

Claim 9: The method of claim 1, wherein after the step (b), the embedded universal integrated circuit card is authenticated by the subscription manager system using at least the digital signature included in the first message.

259. Nix175 teaches "An eUICC subscription manager 109 can take several actions in a step 302b after receiving an identity for the eUICC 107 or module 101. A first action in a step 302b could comprise ***authenticating an eUICC 107*** or a module 101 ***based on the eUICC identity 107a received in the message***. … In addition, the eUICC Subscription manager could ***authenticate using a digital signature algorithm 221***, where the message with the eUICC identity 107a could ***include a digital signature***, as described in the paragraph above." Ex-1016, 49:7-19.

g. Claim 10: The method of claim 1, wherein the decrypted ciphertext is stored in nonvolatile memory of the embedded universal integrated circuit card.

260. Nix175 teaches that the ciphertext decrypted to obtain the "second key K 204" is stored in the eUICC's nonvolatile memory: "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 101 w such that the plaintext second key K 204 remains

available to module 101 after power or a battery 101 k is removed from module 101." Ex-1016, 59:1-5.

h. Claim 11: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module and wherein the module is a machine-to-machine module.

261. Nix175 teaches "In a first exemplary embodiment for a step 308b in FIG. 3 where module 101 supports M2M applications with a M2M service provider 115, MNO 104 could securely send to M2M service provider 115 an identity of module 101, such as, but not limited to module identity 110, eUICC identity 107a, profile identity 107e, and/or the first network module identity 202 through an IP network 111." Ex-1016, 57:14-21.

i. Claim 12: The method of claim 11, wherein the module comprises a module of a type selected from the group consisting of: vending machine, automobile, alarm system, tracking device, monitoring device, security system and remote control device.

262. Nix175 teaches "Among many potential benefits, M2M technologies allow the remote monitoring and/or control of people, assets, or a location where manual monitoring is not economic, or costs can be significantly reduced by using automated monitoring as opposed to manual techniques. Prominent examples today include vending machines, automobiles, alarm systems, and remote sensors." Ex-1016, 1:27-33.

j. Claim 13: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module and wherein the module is a wireless module.

Claim 14: The method of claim 13, wherein the module is configured to communicate using one or more wireless technologies of types selected from the group consisting of: WiFi, WiMax, General Packet Radio Services, Enhanced Data rates for GSM Evolution and 3rd Generation Partnership Project technology.

263. Nix175 teaches "[a]n eUICC 107 within module 101" (Ex-1016, 11:36), where module 101 may operate "as a wireless module" that can communicate with a wireless network using a variety of wireless technologies, include "WiFi, WiMax, a 2nd generation wireless wide area network (WAN) technology such as, but not limited to, General Packet Radio Services (GPRS) or Enhanced Data rates for GSM Evolution (EDGE), [or] 3rd Generation Partnership Project (3GPP) technology such as, but not limited to, 3G, 4G LTE, or 4G LTE Advanced." Ex-1016, 9:45-53.

k. Claim 15: The method of claim 13, wherein the module comprises a module of a type selected from the group consisting of the following: wireless handset, cellular phone, smartphone, laptop computer, tracking device, computer with a radio, circuit board with a radio, hand-held device, netbook, portable computer, multiprocessor system, microprocessor based consumer electronic, programmable consumer electronic, network personal computer, minicomputer, mainframe computer and smartcard.

264. 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.

l. Claim 16: The method of claim 1, wherein the embedded universal integrated circuit card is operably connected to a module, wherein the module is a wired module.

Claim 17: The method of claim 16, wherein the module is configured to communicate using one or more wired connections of types selected from the group consisting of: Ethernet, fiber optic connection and Universal Serial Bus connection.

265. Nix175 teaches "A wireless or a wired configuration for module 101 can be utilized in the present invention." Ex-1016, 9:42-44. When module 101 is a wired module, Nix175 teaches that it can connect to the IP Network "via a wired connection such as, but not limited to, an Ethernet, a fiber optic, or a Universal Serial Bus (USB) connection." Ex-1016, 9:55-58.

m. Claim 18: The method of claim 16, wherein the module comprises a module of a type selected from the group consisting of: computer, security camera, security monitoring device and networked controller.

266. Nix175 teaches "In a wired configuration (not shown), module 101 can be a computer, security camera, security monitoring device, networked controller, etc." Ex-1016, 9:42-44.

n. Claim 19: The method of claim 1, wherein the deriving step (c) is performed prior to the generating step (a).

Claim 20: The method of claim 1, wherein the generating step (a) is performed before the deriving step (c).

267. Nix175 teaches for step (c) that "eUICC private key 215 can be processed or derived using a random number." Ex-1016, 41:63-64. Because the message generated in step (a) is independent of the eUICC private key generation process, it would have been a simple matter of design choice to perform step (c) before step (a) or vice versa.

o. Claim 21: The method of claim 1, wherein the embedded integrated circuit card is operably connected to a module and a user accesses a user interface of the module to enter authentication information before the module receives the symmetric key.

268. Nix175 teaches "the module 101 receives the symmetric key 127 ***after*** the MNO 104 authenticates or verifies that a user 113 or M2N 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.

269. 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.

p. Claim 22: The method of claim 1, wherein the subscription manager receives the ciphertext from a mobile network operator before the network application sends the first message.

270. As discussed above for claim 7, Nix175 teaches "eUICC subscription manager 109 could assemble and cipher the profile 107d into a profile 107c at an earlier time, such as before receiving the eUICC identity 107a." Ex-1016, 50:57-59.

XIII. CONCLUSION

271. For the reasons above, it is my opinion that Claims 1-22 of the '094 Patent are obvious based on each of the grounds specified in this Declaration.

272. 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