UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
61/762,098	02/07/2013		570	0076412-000129		

CONFIRMATION NO. 1099

FILING RECEIPT

21839
BUCHANAN, INGERSOLL & ROONEY PC
POST OFFICE BOX 1404
ALEXANDRIA, VA 22313-1404

OC000000059348714

Date Mailed: 02/20/2013

Receipt is acknowledged of this provisional patent application. It will not be examined for patentability and will become abandoned not later than twelve months after its filing date. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Mehdi COLLINGE, Braine-l'Alleud, BELGIUM;
Susan Thompson, Burland, UNITED KINGDOM;
Patrik Smets, Nijlen, BELGIUM;
David Anthony Roberts, Appleton, UNITED KINGDOM;
Simon E.J. Phillips, York, UNITED KINGDOM;

Applicant(s)

MasterCard International Incorporated, Purchase, NY

Power of Attorney:
Charles Wieland III--33096

Permission to Access - A proper **Authorization to Permit Access to Application by Participating Offices** (PTO/SB/39 or its equivalent) has been received by the USPTO.

If Required, Foreign Filing License Granted: 02/14/2013

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 61/762,098**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No

IPR2025-01152
Apple EX1008 Page 1

Title

SYSTEMS AND METHODS FOR PROCESSING MOBILE PAYMENTS BY PROVISIONING CREDENTIALS TO MOBILE DEVICES WITHOUT SECURE ELEMENTS

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The

IPR2025-01152
Apple EX1008 Page 2

date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

APPLICATION FOR
UNITED STATES LETTERS PATENT

FOR

SYSTEMS AND METHODS FOR PROCESSING MOBILE PAYMENTS BY PROVISIONING CREDENTIALS TO MOBILE DEVICES WITHOUT SECURE ELEMENTS

By

Mehdi Collinge

Susan Thompson

Patrik Smets

David Anthony Roberts

and

Simon Phillips

Attorney Docket No. 0076412-000129
BUCHANAN INGERSOLL & ROONEY PC
CUSTOMER NO. 21839
P.O. Box 1404
Alexandria, VA 22313-1404

Buchanan Ingersoll & Rooney PC
Attorneys & Government Relations Professionals

SYSTEMS AND METHODS FOR PROCESSING MOBILE PAYMENTS BY PROVISIONING CREDENTIALS TO MOBILE DEVICES WITHOUT SECURE ELEMENTS

RELATED APPLICATIONS

[0001] This application claims the priority benefit of commonly assigned U.S. Provisional Application No. 61/619,095 filed April 2, 2012, for "Systems and Methods for Processing Mobile Payments for Mobile Devices Without Secure Elements," by Simon Phillips et al., U.S. Provisional Application No. 61/639,248 filed April 18, 2012, for "Systems and Methods for Processing Mobile Payments By Provisioning Credentials to Mobile Devices Without Secure Elements," by Simon Phillips et al., and U.S. Provisional Application No. 61/735,383 filed December 10, 2012, for "Systems and Methods for Processing Mobile Payments by Provisioning Credentials to Mobile Devices Without Secure Elements," by Mehdi Collinge et al., all of which are each herein incorporated by reference in their entirety.

FIELD OF THE DISCLOSURE

[0002] The present disclosure is directed to a method and system providing technical solutions for processing electronic payments initiated from a mobile device without requiring a secure element (SE) in the mobile device using in part a financial transaction card processing system or network as a part thereof.

BACKGROUND

[0003] Advances in mobile and communication technologies have created tremendous opportunities, one of which is providing users of mobile computing devices an ability to initiate payment transactions using their mobile device. One approach to enable mobile devices to conduct payment transactions is through the use of near field communication

(NFC) technology to securely transmit payment information to a contactless terminal. To enable this, mobile phones with secure element hardware (i.e., a secure element chip) can be used to securely store payment account credentials, such as credit card credentials, have been used. Additionally, the use of mobile devices configured to operate with a PayPass® chip have been proposed. However, not all mobile phones have secure elements. Additionally, not all issuers, acquirers or merchants have host systems that can process chip data elements. Lastly, some issuers cannot access a secure element (even if one is available). As a result, a user who has an NFC-capable mobile device may not be able to use it as a payment device if their mobile device lacks a secure element (SE) and even in some cases where their mobile device has an SE.

[0004] Accordingly, what are needed are systems and methods that provide technical solutions to allow mobile devices without an SE to complete contactless payments. What is further needed are systems and methods that allow mobile devices without an SE to complete mobile payments using a Cloud-based transaction data generation system which can authenticate and generate payment credentials (i.e., tokens) associated with one or more existing payment accounts, such as, but not limited to, a PayPass® account, so that the user can conduct PayPass® transactions at PayPass®-enabled merchants with a mobile device without having to use an SE and without requiring their acquirer or merchant to make significant changes to their host system(s).

SUMMARY

[0005] Methods and systems are disclosed for enabling payments via a mobile device, such as a smartphone, without requiring use of a secure element (SE) on the mobile device.

[0006] According to an embodiment, a set of processes deliver solutions for contactless payments, such as online transactions at a Point-of-Sale (POS), when using a mobile device but not requiring use or presence of an SE. One embodiment uses a combination of remote authentication and the provisioning of payment credentials to the mobile device for one transaction. In an alternative embodiment, remote notification is performed and payment

credentials are provisioned to a mobile device without an SE for a limited number of transactions.

[0007] In yet another embodiment, payments from mobile devices are processed using an available Trusted Execution Environment (TEE) and the TEE hosts services used during the authentication and payment processing in conjunction with a Mobile Authentication Application (MAA), without requiring use of an SE. Alternatively, payments can be processed when a TEE is not available using secure storage combined with camouflaging data by using a unique Mobile Device ID as a parameter along with the Personal Value.

[0008] According to an embodiment, a contactless payment process between a mobile device and a POS is processed as a standard payment transaction and does not require significant updates to transaction acquirer or merchant systems.

[0009] Certain exemplary embodiments provide a trusted environment for mobile authentication and/or mobile payment services even when mobile devices lacking or not using an SE are used to initiate payments. An exemplary embodiment disclosed herein uses a non-SE based solution to support mobile authentication by employing an MAA.

[0010] A high level flow for an exemplary embodiment of the process flow begins with provisioning of authentication credentials to a mobile device (i.e., at the edge of a payment processing network), which can subsequently be used for authentication based upon accessing locally stored credentials on the mobile device. Next, payment credentials are accessed from the Cloud (i.e., a Cloud-based transaction data generation system) so that remote identification and authentication can be performed based upon the credentials stored in the Cloud. Then, authentication is performed using an MAA. According to one embodiment, an MAA is a software implementation of MasterCard Authentication Solutions (i.e., two-factor authentication using a Chip Authentication Program (CAP) Token). As used herein, in an embodiment, the CAP Token can be conceptualized as a dynamic One-Time Password (OTP) that cannot be reused. Both CAP and PLA (perso-less authentication) technologies use a CAP Token to support the authentication process. PLA technology is discussed in further detail in WIPO Published Application No. 2010/030362, published March 18, 2010, to Collinge et al., which is herein incorporated by reference in its entirety.

CAP technology is discussed in further detail in WIPO Published Application No. 2005/001618, published January 6, 2005, to Rutherford et al., which is herein incorporated by reference in its entirety.

[0011] In accordance with another exemplary embodiment, mobile authentication and mobile payment services are implemented as an online-only solution wherein a CAP token is verified online by a CAP Token Validation Service (CTVS). According to this embodiment, a Personal Value, gesture, or passcode is used to retrieve a valid attribute, such as an AC_{CMK} key, which may be used to generate an Application Cryptogram (AC). The solution further includes a wrong key detection mechanism (as result of wrong Personal Value or passcode tries). In one embodiment, the wrong key detection is supported by an issuer (e.g., CAP Token validation failure). Advantageously, the solution does not persistently or permanently store any additional sensitive assets, such as a primary account number (PAN) in the MAA. In an embodiment, a PAN is not stored for authentication services, but some payment credentials are stored for a limited time, including track data. These payment credentials are protected using a storage key, but they contain PAN information. In one embodiment, a protocol is defined in order to avoid any disclosure of complete AC values. Another advantage of solution is that it uses secure coding best practices (e.g., rules for management of sensitive assets such as a Personal Value and/or temporary values such as a generated AC).

[0012] In order to prevent cloning of camouflaged data, an embodiment uses a unique Mobile Device ID as a parameter along with the Personal Value.

[0013] As a consequence, the non-SE based solution disclosed herein can be used for mobile authentication services, including services that access payment credentials stored in the Cloud.

[0014] According to another exemplary embodiment, a solution provides architecture for completing a two-step process for remote authentication and remote payment. This solution overcomes the lack of access to a trusted environment from a mobile device without impacting the security level of the architecture, even when payment credentials are stored in the MAA to support proximity payment when there is no connectivity to the Cloud (i.e.,

when a mobile device momentarily lacks Wi-Fi/ General packet radio service (GPRS) network connectivity).

[0015] In yet another embodiment, a set of processes delivers a payment token payload to a mobile application installed on a mobile device. According to this embodiment, information (supported by functions) can be used to process contactless payments (online transactions at a POS) using PayPass® magnetic stripe (PayPass® Magstripe) transactions without using a requiring a Secure Element. This 'push model' embodiment uses a combination of several mechanisms to support a registration process, an initialization process and a provisioning process after remote notification. Security mechanisms are used to protect the payment credentials, to deliver an encrypted payment token payload that can only be used for one transaction, and to mitigate risk from mobile cloning. The push model does not require changes to the acceptance environment and supports PayPass® Magstripe transactions without any connection to the Cloud at time of the transaction with a storage of a limited set of pre-computed payment credentials.

[0016] Exemplary methods for provisioning payment account credentials from a Cloud-based system using a "mobile cloud account" to an NFC-enabled mobile device on behalf of an issuer are described in U.S. Provisional Application Serial No. 61/605,588 entitled "Systems and Methods For Mapping a Mobile Cloud Account to a Payment Account," filed on March 1, 2012, the disclosure of which is hereby incorporated by reference in its entirety.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] Figure 1 is a diagram of an exemplary system in which a Mobile Authentication Application (MAA) can be used to process an electronic payment from a mobile device without requiring a secure element (SE), in accordance with an exemplary embodiment of the present disclosure.

[0018] Figures 2A and 2B are storyboards depicting provisioning processes for downloading, installing, provisioning, activating and using a mobile payment application with a mobile computing device, in accordance with exemplary embodiments of the present disclosure.

[0019] Figure 3 is a diagram of a system illustrating a high level process flow between system components for completing a contactless payment from a mobile computing device without requiring an SE, in accordance with an exemplary embodiment of the present disclosure.

[0020] Figure 4 is a diagram depicting components of a system for completing a contactless payment from a mobile computing device without requiring an SE, in accordance with an exemplary embodiment of the present disclosure.

[0021] Figure 5 illustrates a process flow for provisioning payment credentials for a contactless payment, in accordance with an exemplary embodiment of the present disclosure.

[0022] Figure 6 illustrates a process flow for processing a contactless payment transaction, in accordance with an exemplary embodiment of the present disclosure.

[0023] Figure 7 illustrates a process flow for approving a contactless payment transaction, in accordance with an exemplary embodiment of the present disclosure.

[0024] Figure 8 is a detailed diagram of an exemplary system illustrating the process flow for completing a contactless payment from a mobile computing device without requiring an SE, in accordance with an exemplary embodiment of the present disclosure.

[0025] Figure 9A is a diagram of a system illustrating a pull model process with high level flows between system components for completing a contactless payment from a mobile computing device without requiring an SE, in accordance with an exemplary embodiment of the present disclosure.

[0026] Figure 9B is a diagram of a system illustrating a push model process with high level flows between system components for completing a contactless payment from a mobile computing device without requiring an SE, in accordance with an exemplary embodiment of the present disclosure.

[0027] Figure 10 illustrates a communications sequence for registering a consumer/user and providing an activation code for an initialization process as part of a push model, in accordance with an exemplary embodiment of the present disclosure.

[0028] Figure 11 illustrates a communications sequence for activating an account and service, defining an access code, and initializing a mobile application as part of a push model, in accordance with an exemplary embodiment of the present disclosure.

[0029] Figures 12A and 12B illustrate a communications sequence for a (first) provisioning of payment credentials as part of a push model, in accordance with an exemplary embodiment of the present disclosure.

[0030] Figure 13 is a diagram of a system illustrating flows between system components to retrieve an encrypted payment token payload after notification as part of a (first) provisioning of payment credentials for a push model, in accordance with an exemplary embodiment of the present disclosure.

[0031] Figure 14 illustrates a communications sequence for using a mobile payment application to access a locally stored encrypted payment token payload to process a standard PayPass® Magstripe transaction as part of a push model, in accordance with an exemplary embodiment of the present disclosure.

[0032] Figure 15 is a diagram of a system illustrating flows between system components to use a mobile payment application to access a locally stored encrypted payment token payload to process a standard PayPass® Magstripe payment transaction as part of a push model, in accordance with an exemplary embodiment of the present disclosure.

[0033] Figure 16 depicts an example computer system in which embodiments of the present invention may be implemented.

[0034] Figure 17 is a diagram of a system depicting components of an alternative system for stored payment token payloads in a mobile application in accordance with an exemplary embodiment of the present disclosure.

[0035] Figure 18 illustrates a process flow for dual channel communication between a payment credentials management server and a mobile application in accordance with an exemplary embodiment of the present disclosure.

[0036] Figure 19 is a diagram of an alternative system using the components of Figure 17 in which a mobile application can be used to process an electronic payment from a mobile

device without requiring a secure element in accordance with an exemplary embodiment of the present disclosure.

[0037] Figure 20 illustrates a process flow for delivering a single use key from a payment credentials management server to a mobile application program on a mobile device for use in a contactless payment transaction in accordance with an exemplary embodiment of the present disclosure.

[0038] The features and advantages of the present disclosure will become more apparent from the detailed description set forth below when taken in conjunction with the drawings, in which like reference characters identify corresponding elements throughout. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. Generally, the drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number.

DETAILED DESCRIPTION

[0039] As used herein, "payment account", “credit card number" and "credit card" are sometimes used interchangeably. These terms mean a credit card, debit card, pre-paid card, hybrid card, plastic or virtual card number (VCN)(single use, limited use or simply virtual), or nearly any other account number that facilitates a financial transaction using a transaction clearance system. VCNs and pre-paid card numbers and other financial transaction card number that can be generally viewed as being more readily issued and disposed of because they do not require the establishment of a line of credit, and optionally can be linked to various controls (amounts, cumulative amounts, duration, controls on spending by amounts, cumulative amounts, types of merchants, geographic controls, to name a few). As used herein, these types of cards (VCN, pre-paid, etc.) are sometimes referred to as intelligent transaction card (ITC) numbers. As used herein, the term "payment account" is sometimes used interchangeably with a payment account number and means a credit card, the account number for a credit card, or any identifier that can be used to link a payment account to a purchase transaction initiated from a mobile device.

[0040] As used herein, the terms "user", "customer", "consumer", "account holder", "cardholder", and "card user" can be used interchangeably and can include any user making purchases of goods and/or services. Unless specifically stated differently or from context, in exemplary embodiments, a user may be interchangeably used herein to identify a human customer, a software application, or a group of customers and/or software applications executed by one or more consumers to conduct a purchase transaction. Besides a human customer who can purchase items using a mobile device, a software application can be used to process purchases. Accordingly, unless specifically stated, the terms "user", "customer", "cardholder", "account user" and "card user" as used herein do not necessarily pertain to a human being.

[0041] Further, as used herein, the term "issuer" can include, for example, a financial institution (e.g., bank) issuing a card, a merchant issuing a merchant specific card, a stand-in processor configured to act on-behalf of the card-issuer, or any other suitable institution configured to issue a financial card. Finally, as used herein, the term "transaction acquirer" can include, for example, a merchant, a merchant terminal, a point-of-sale (POS) terminal at a merchant, or any other suitable institution or device configured to initiate a financial transaction per the request of a customer.

[0042] Exemplary phone-based electronic wallets capable of providing authenticated transactions across multiple channels of commerce are described in U.S. Application No. 13/209,312, entitled "Multi-Commerce Channel Wallet for Authenticated Transactions," filed on August 12, 2011, which claims the benefit of U.S. Provisional Application Serial No. 61/372,955 filed August 12, 2010 and U.S. Provisional Application Serial No. 61/468,847 filed March 29, 2011, the disclosures of which are hereby incorporated by reference in their entireties.

[0043] Identification of PayPass® Magstripe transactions performed using the solution described below with reference to Figures 3-8, 9A, 9B and 10-15 may require identification of a primary account number (PAN) using, for example, a specific range (and/or BIN). Examples of systems and methods for routing electronic transactions through financial processing systems (e.g., debit/credit networks) as a part of an electronic payment system are

described in U.S. Application No. 13/078,374, entitled "Method for Performing Acquirer Routing and Priority Routing of Transactions," filed on April 1, 2011, which is incorporated herein by reference in its entirety.

I. Exemplary System Embodiment

[0044] Figure 1 is a block diagram of an exemplary system 100 for processing an electronic payment initiated by a mobile device without an SE, according to exemplary embodiments of the present disclosure. As implemented in the presently described exemplary embodiment, the system 100 depicted in Figure 1 includes a mobile device 104 without an SE, a point of sale (POS) terminal 181, a payment processor 103 (e.g., MasterCard) with a payment processing network 170 (e.g., MasterCard's Worldwide Network) that facilitates routing of mobile payment transactions for authorization, a mobile authentication application (MAA) 111, a transaction acquirer 166, and an issuer 180. As will be appreciated by those skilled in the relevant art(s), while the exemplary POS terminal 181 is depicted as a MasterCard PayPass® terminal, other contactless POS terminals 181 with NFC capabilities can be used.

[0045] The system 100 performs authentication using the MAA 111 as part of a pull model. According to an embodiment, the MAA 111 is a software implementation of MasterCard Authentication Solutions (two-factor authentication using a CAP Token). A CAP Token Generation Service (CTGS) can be integrated in a mobile application to build a MasterCard Authentication Solution for mobile device 104 where the cardholder 113 uses the Mobile Authentication Application (MAA) to generate a CAP Token.

[0046] Although the MAA 111 is depicted in Figure 1 as being hosted by the mobile device 104, it is to be understood that in alternative embodiments, the MAA 111 can be hosted by the issuer 180 or a third party such as transaction processors. As used herein, in an embodiment, a transaction is distinguished from an authentication transaction, which is used to get access to payment credentials managed in the cloud-based transaction data generation system) and the payment transaction (i.e., a standard PayPass® Magstripe transaction). For example, it should be understood that the MAA 111 can alternatively be external to the payment processor 103. By way of example and not limitation, in one embodiment, the MAA 111 can reside on a computing device associated with the issuer 180. According to

embodiments, generation of the CAP Token can be done by the MAA 111 component of the mobile payment application or might be done using another form factor. Examples of third party transaction processors that may utilize the MAA 111 include, but are not limited to, outsourced transaction processors such as PrePaid Services (PPS), ElectraCard Services (ECS), First Data Resources (FDR), and providers of mobile wallet applications such as the MasterCard wallet. Examples of such mobile wallet applications capable of providing authenticated transactions across multiple channels of commerce are described in U.S. Application No. 13/209,312, entitled "Multi-Commerce Channel Wallet for Authenticated Transactions," filed on August 12, 2011, which claims the benefit of U.S. Provisional Application Serial No. 61/372,955 filed August 12, 2010 and U.S. Provisional Application Serial No. 61/468,847 filed March 29, 2011, the disclosures of which are hereby incorporated by reference in their entireties.

[0047] The system 100 allows a user 113 to use nearly any mobile computing device 104 having near field communications (NFC) capabilities to make purchases with a payment account, including, but not limited to, a Personal Digital Assistant (PDA), a tablet computing device, an iPhone™, an iPod™, an iPad™, a device operating the Android operating system (OS) from Google Inc., a device running the Microsoft Windows® Mobile OS, a device running the Microsoft Windows® Phone OS, a device running the Symbian OS, a device running the webOS from Hewlett Packard, Inc., a mobile phone, a BlackBerry® device, a smartphone, a hand held computer, a netbook computer, a palmtop computer, a laptop computer, an ultra-mobile PC, a portable gaming system, or another similar type of mobile computing device having a capability to make electronic purchases using a payment account (i.e., credit card).

[0048] With reference to Figure 1, the payment processor 103, provide various services and product offerings to support customers and vendors. In one embodiment, the payment processor 103 can use the MasterCard Internet Service, which includes the InControl™ product offering. Examples of such product offerings are described in U.S. Patent No. 6,315,193; U.S. Patent No. 6,793,131; U.S. Application No. 10/914,766, filed on August 9, 2004; U.S. Application No. 11/560,112, filed on November 15, 2006; U.S. Application No.

12/219,952, filed on July 30, 2008; and International Application No. PCT/US2009/002029, filed on September 19, 2009, U.S. Published Patent Applicaton No. 2009/0037333, filed on July 30, 2008, all incorporated herein by reference in their entirety (herein the controlled payment numbers or CPN Patents).

[0049] The communication links depicted in the system 100 between the various components can be through public and/or private networks or virtual private networks (e.g., the Internet and mobile networks particularly with respect to communications with mobile device 104, and private networks such as payment processing network 170).

[0050] As shown in Figure 1, system 100 processes the payment by user 113 at the POS terminal 181 using a standard process for the payment transaction using a transaction acquirer 166, a payment processor 103, and an issuer 180.

[0051] The processing for a payment in system 100 begins when a transaction is initiated by a user 113 with a mobile device 104 at a POS terminal 181. As illustrated in Figure 1, the mobile device 104 does not have a secure element (SE).

[0052] Authentication to the Cloud is performed within system 100 in order to retrieve an encrypted payload 112 from a cloud-based transaction data generation system 106 to the mobile device 104. The cloud-based transaction data generation system 106 comprises a transaction data generation service 108 that is configured to generate payment tokens and other data needed to complete purchases using the mobile device 104. As shown in Figure 1, the cloud-based transaction data generation system 106 further comprises key storage 110 for storing keys and encrypted information 179. In the exemplary embodiment of Figure 1, the encrypted information 179 has been encrypted using $K_{Storage}$ and includes Track 1 data, and/or Track 2 data. As shown in Figure 1 the encrypted payload 112 is provisioned to the mobile device 104 from the cloud-based transaction data generation system 106. The cloud-based transaction data generation system 106 also includes a payment credentials management system 114 and an authentication service 116. In the exemplary embodiment of Figure 1, the authentication service 116 is configured to perform CAP Token validation (CTVS) and can use a Chip Authentication Program (CAP) token for authentication.

[0053] As described below with reference to Figures 2A and 2B, a cardholder (i.e., a user) 113 can provision a mobile payment application to the mobile device 104.

[0054] Next, an authorization request 168 is submitted. As shown in Figures 3-8, the encrypted payload 112 is not sent to the acquirer 166 or a merchant. The mobile payment application uses the content of the encrypted payload 112 to perform the PayPass® Magstripe transaction. This can be done using information from a PayPass® reader, such as, but not limited to, an UN_{Reader}.

[0055] The acquirer 166 then routes authorization request 168 to a payment processing network 170 associated with the payment processor 103 (e.g., MasterCard).

[0056] Based on information contained in the authorization request 168, using at least the included content of the encrypted payload 112, payment credentials 174 are generated and provisioned to the MAA 111 in the case of the pull model.

[0057] At this point, the payment process is done by the mobile payment application by using the payment credentials 174 in a payment credentials management system 114. As shown in Figure 1, it is to be understood that the payment credentials management system 114 is connected to the issuer 180.

[0058] The mobile payment application may generate a cryptogram. This cryptogram may be forwarded with the authorization request 168 to the acquirer 166. As shown in Figure 1, this can be further sent to the payment processing network 170. In an embodiment, the cryptogram 178 may be generated using key management services (i.e., through CVC3 validation, including dynamic CVC3 validation).

[0059] The payment processor 103 then routes an authorization request 168 based on the payment credentials 174 and the cryptogram 178 to the issuer 180 and the issuer 180 responds to the authorization request 168 with the authorization response 172. In one embodiment, system 100 includes a connection 178 between the issuer 180 and a payment credentials management system 114.

[0060] After receiving the authorization response 172, the payment processor 103 forwards the authorization response 172 to the acquirer 166, which in turn routes the authorization response 172 back to the POS terminal 181.

II. Mobile Payment Application Provisioning Methods

[0061] Figures 2A and 2B are storyboards depicting provisioning processes 200 and 220, respectively, for downloading, installing, provisioning, activating, and using a mobile payment application with a mobile computing device 104, in accordance with exemplary embodiments of the present disclosure. Figures 2A and 2B are described with continued reference to the embodiment illustrated in Figure 1. However, Figures 2A and 2B are not limited to that embodiment.

Mobile Payment Application Provisioning using a Pull Model

[0062] Figure 2A depicts provisioning process 200 for downloading, installing, provisioning, activating, and using a mobile payment application with a mobile computing device 104 using a 'pull model,' in accordance with an embodiment of the present disclosure.

[0063] With reference to Figure 2A, in step 201 a user registration process is completed. As shown in Figure 2A, this step can be accomplished using input supplied by a cardholder or user 113 via a GUI 202 of the user's 113 mobile device 104.

[0064] With continued reference to Figure 2A, in step 203, the mobile payment application is downloaded and installed.

[0065] In step 205, authentication credentials associated with a payment card are provisioned to the mobile device 104.

[0066] In step 207, the mobile device 104 is authenticated to the Cloud-based transaction data generation system 106 in order to retrieve payment credential, such as, but not limited to tokens. As shown in Figure 2A, this step comprises synchronization between the Cloud-based transaction data generation system 106 and the issuer 180 systems.

[0067] In step 209, the mobile payment application is activated. As shown in Figure 2A, this step can be accomplished using a mobile payment application to activate a contactless interface in order to enable a contactless payment using the mobile device 104 to make a payment at a POS terminal 181. For example, step 209 can comprise activating an NFC interface using the mobile payment application. Payment credentials, such as, but not limited to tokens, can then be redeemed at the POS terminal 181 to make a payment using the mobile payment application. As used herein, in an embodiment, this redemption refers to

processing the information from the cloud-based transaction data generation system 106 to make the payment. In this step, the cloud-based transaction data generation system 106 looks at credentials/transaction tokens stored in a mobile device 104 (i.e., the smart phone depicted in Figure 2A).

[0068] In step 211, the mobile device 104 is ready for a subsequent, next payment (i.e., by repeating step 209, or can be used to retrieve additional payment credentials by returning control to step 207.

Mobile Payment Application Provisioning using a Push Model

[0069] Figure 2B depicts provisioning process 220 for downloading, installing, provisioning, activating, and using a mobile payment application with a mobile computing device 104 using a 'push model,' in accordance with an embodiment of the present disclosure.

[0070] Figure 2B depicts a registration process to allow user registration to a contactless payment service (SE-less Mobile PayPass). As shown in Figure 2B, an initialization process lets a user 113 perform activation of the service, define an access code, and initialize a mobile payment application. Process 220 includes a provisioning process supported by a remote notification service (see the remote notification service 915 described with reference to Figure 9B below).

[0071] In an embodiment, the process 220 also uses online access to the Cloud-based transaction data generation system 106 at the time of the provisioning to allow retrieval of a new, encrypted payment token payload. The process 220 is integrated with the Cloud-based transaction data generation system 106 to enable interaction with the payment credentials Management system 114, to interface with the remote notification service 915, and to complete a synchronization process with the issuer 180 systems. According to an embodiment, both of processes 200 (pull model) and 220 (push model) can include updating the CVC3 (Dynamic Card Validation Code) validation process, which can be carried out by the issuer 180.

[0072] In accordance with an embodiment, the process 220 provides access from the mobile payment application to an NFC interface of the mobile device 104 to support a contactless

payment transaction (i.e., a PayPass® Magstripe transaction) with a suitable reader (i.e., a PayPass® reader).

[0073] As shown in Figure 2B, in step 201 a user registration process is completed. As shown in Figure 2B, this step can be accomplished using input supplied by a cardholder or user 113 via a GUI 202 of the user's 113 mobile device 104.

[0074] With continued reference to Figure 2B, in step 203, the mobile payment application is downloaded and installed.

[0075] In step 204, the mobile payment application for the mobile device 104 is initialized. As shown in Figure 2B, this step can comprise defining an access code.

[0076] In step 206, the user 113 of the mobile device 104 is notified that payment credentials, such as, but not limited to tokens, can be retrieved from the Cloud-based transaction data generation system 106. As shown in Figure 2B, this step comprises using a provisioning process. In the embodiment depicted in Figure 2B, the provisioning process of step 206 may require an access code to load credentials and synchronization between the Cloud-based transaction data generation system 106 and the issuer 180 systems.

[0077] In step 208, the mobile payment application is activated. As shown in Figure 2B, this step can be accomplished using a mobile payment application to activate a contactless interface in order to enable a contactless payment using the mobile device 104 to make a payment at a POS terminal 181.

[0078] For example, step 208 can comprise activating an NFC interface using the mobile payment application. Payment credentials, such as, but not limited to tokens, can be redeemed at the POS terminal 181 to make a payment using the mobile payment application. As shown in Figure 2B, in this step, an access code can be required in order to use the credentials. For example, the cloud-based transaction data generation system 106 will look at credentials/transaction tokens stored in a mobile device 104 (i.e., the smart phone depicted in Figure 2B) after the access code has been furnished. According to this embodiment, process 220 provides a proof of knowledge of the access code without sending this value to the cloud-based transaction data generation system 106.

[0079] In step 210, the mobile device 104 is ready for a subsequent, next payment (i.e., by repeating step 208, or can be used to retrieve additional payment credentials by returning control to step 206. As shown in Figure 2B, in an optional embodiment, step 210 can comprise notifying the user 113 when a PayPass® transaction has been performed.

[0080] Exemplary processes for authentication are described below for the pull model. Exemplary payment, and synchronization processes are also described below for both the pull and push models.

Authentication Process for the Pull Model using the MAA

[0081] In accordance with an embodiment, the principles for the authentication process (mobile device 104 to the Cloud-based transaction data generation system 106) are:

1. Integrate MasterCard MAA solution (with a CAP Token) in the SE-less Mobile Payment Application to generate a CAP Token to support the authentication process.
2. Any use of MAA assumes the availability of a process to install the MAA component and provision it with a Virtual Card Profile used for authentication purposes. The Virtual Card Profile is associated with a Payment Card (Payment Credentials).
3. Access control must be defined to grant access to the assets of MAA (e.g. protected using some mechanisms such as camouflage). According to an embodiment, an online PIN value cannot be used to grant access to MAA and the generation of a valid CAP Token. In accordance with this embodiment, a gesture or a password must be used instead.
4. If the validation of the CAP Token is successful, the cloud-based transaction data generation system 106 generates the CVC3 value using a genuine KD_{CVC3} and returns an encrypted payload to the Mobile Payment Application.
5. If the validation of the CAP Token is NOT successful, the Cloud-based transaction data generation system 106 generates CVC3 using a 'fake' KD_{CVC3} and returns an encrypted payload (see, e.g., encrypted payload 112 depicted in Figures 3-8, 9A and 9B) to the Mobile Payment Application. At the same time an alarm is triggered to the issuer 180.

Synchronization Process

[0082] According to an embodiment, the principles for the synchronization process (between the Cloud-based transaction data generation system 106 to issuer 180) are as follows:

1. The Cloud_CVC3$_{TRACK1/2}$ generation is managed in the Cloud-based transaction data generation system 106 ("the Cloud). This encompasses the generation of KS_{UN} and UN_{CLOUD}, and the application transaction counter (ATC) management.
2. (KS_{UN}, Cloud_CVC3$_{TRACK1/2}$ and ATC) are returned using an encrypted Payload to the Mobile Payment Application.
3. (UN_{CLOUD} and ATC) [Including optionally some status information] are sent to the Issuer.
4. The Issuer has a means (e.g. Using PAN or information available in the Payment Transaction) to identify transactions that require additional processing for the retrieval of the UN_{CLOUD} and KS_{UN} values using the ATC value provided in that Payment Transaction.

Payment Process

[0083] In accordance with an embodiment, the principles for the payment process (mobile device 104 to the Cloud-based transaction data generation system 106) include the following:

1. The Mobile Payment Application must have retrieved at least one (KS_{UN}, Cloud_CVC3$_{TRACK1/2}$, ATC) before the Tap.
2. The dynamic values (CVC3 and ATC) are used as a first form factor to authenticate the payment transaction. The Online PIN can be used as a second form factor. This dynamic CVC3 value is generated by the mobile payment application using information from the payload provided by the cloud-based transaction data generation system 106 ("the Cloud").

III. Exemplary Authentication and Transaction Process Flows

[0084] Figures 3-8, 9A, 9B, 13 and 15 are diagrams of the system 100 illustrating data flows for authentication and transactions used to process contactless payments from a mobile computing device without requiring an SE. Figures 3-8, 9A, 9B, 13 and 15 depict varying

levels of detail for data and process flows for contactless payments that do not require use of an SE.

Pull Model Flow

[0085] Figures 3-8 and 9A depict data flows within the system 100 using a pull model. Figures 3-8 and 9A are described with continued reference to the embodiments illustrated in Figures 1 and 2A. However, Figures 3-8 and 9A are not limited to those embodiments.

[0086] As shown in Figure 3, authentication to the cloud-based transaction data generation system 106 is performed to retrieve the payment credentials 174.

[0087] System 100 includes an authentication module configured to perform authentication of a user 113 based on information the user 113 knows (i.e., the user authentication 318 depicted in Figure 3). In embodiments, the authentication module can use a user ID or account number in conjunction with other information the user 113 knows, such as passcode, gesture or other suitable Personal Value. As shown in Figure 8, the user authentication 318 is performed separately from the remote authentication 824 of payment credentials.

[0088] After the authentication module authenticates the user 113, the user 113, who in the exemplary embodiment of system 100 is depicted as a cardholder, initiates shopping by making a selection 304 of one or more items to place in a shopping cart 306. As would be understood by persons skilled in the relevant art, selection 304 and shopping cart 306 can be performed at 'brick and mortar' merchants at a POS, with payments for items in shopping cart 306 being made via a proximity payment. As shown in FIG. 3, the system 100 routes a payment request 307 to the merchant's POS terminal 181.

[0089] Exemplary data flows for the reference numerals 1-20 and labels A-C depicted in Figures 5-7 are described in Table 1 below.

Table 1

Reference	Description
A1	Embodiments can support several levels of authentication: • The Access to the Mobile Device (e.g. Device Locking mechanism) • The Access to the Mobile Payment Application • The Access to the cloud-based transaction data generation system 106 ("Cloud system")
A2	The Authentication component (using MAA technology) has to be provisioned
B	At time of the authentication credentials provisioning, a storage key is also stored in the Mobile Payment Application. This key is used to protect the static payment credentials and the transport of the payload from the Cloud to the Mobile Payment Application
C	At time of the authentication credentials provisioning, static payment credentials are also provisioned
1	The Cardholder uses a SE-less Mobile PayPass® Payment Application
2a	The Cardholder connects to the Cloud to retrieve Payment credentials
2b	The Cardholder uses the MAA component of the Mobile Payment Application to generate a CAP Token for the authentication transaction. The Cardholder has to supply some credentials (e.g. A gesture, a password…)
3	Mobile Payment Application sends a CAP Token to the Cloud

Table 1

Reference	Description
4	The Payment System (in the Cloud) validates the CAP Token using a CAP Token Validation Service (CTVS). The Payment System can be operated by MasterCard or by the Issuer.
5	The CTVS validates the CAP Token. The CTVS can be operated by MasterCard or by the Issuer.
6	The result of the CAP Token validation is sent to the Payment Credentials Management System. The Payment Credentials Management System can be operated by MasterCard or by the Issuer.
7	Upon successful authentication, a genuine KD_{CVC3} is used and (KS_{UN}, $Cloud\_CVC3_{TRACK1/2}$, ATC) is returned. Upon unsuccessful authentication, a fake KD_{CVC3} key is used and (KS_{UN}, $Cloud\_CVC3_{TRACK1/2}$, ATC) is returned and an alarm is triggered.
8a	There is synchronization process between the Cloud and the Issuer. The synchronization process may include the definition of rules for the validity of the generated CVC3 values.
8b	(KS_{UN}, $Cloud\_CVC3_{TRACK1/2}$, ATC) is returned to the front-end of the Cloud system for delivery to the Mobile Payment Application.
9	(KS_{UN}, $Cloud\_CVC3_{TRACK1/2}$, ATC) is returned to the Mobile Payment Application. This can encompass additional payment assets.

Table 1

Reference	Description
10	(KS_{UN}, Cloud_$CVC3_{TRACK1/2}$, ATC) and the additional assets are stored. The Mobile Payment Application is ready to support a PayPass® Magstripe Payment using a Mobile Device
11	Standard shopping experience.
12	Standard PayPass® Magstripe payment experience using a Mobile Device. The Mobile Payment Application use a specific process to generate the CVC3 using (KS_{UN}, Cloud_$CVC3_{TRACK1/2}$, ATC and UN_{READER})
13	The Cardholder may need to enter the Online PIN at the POS (using a PED).
14	The PayPass® Terminal executes the standard payment transaction process.
15	A standard payment transaction authorization message is used. It contains the UN from the PayPass® Reader ([Partial Info] UN_{READER}), the CVC3 and the ATC [Partial Info] provided by the Mobile Payment Application. It contains the PIN Block when Online PIN is used.
16	A standard Online PIN translation process can take place between the Acquiring environment and the Issuing environment.
17	The standard processes are used for the Payment Transaction.
18	The standard Online PIN verification process applies (if applicable)

Table 1

Reference	Description
19a	The Issuer has a mean to identify transaction that requires additional processing for CVC3 validation when an embodiment using SE-less Mobile Contactless Payment is used. Using the ATC provided in the Payment Transaction, the Issuer is able to retrieve the UN_{CLOUD} and KS_{UN} values that were used by the Payment Credential Management System to generate the CVC3 value. Detection of unsuccessful authentication can also take place at this stage.
19b	A standard process applies for the CVC3 validation using the UN_{CLOUD} and KS_{UN} values.
20	The completion of the Payment transaction process remains unchanged.

[0090] As shown in Figures 3-8, generation of the payment credentials 174 and provisioning of the payment credentials 174 to the MAA 111 is performed as part of the mobile payment processing flow. In system 100, this can be accomplished by using an Application Cryptogram (AC) for an authentication process and using the CVC3 keys 118 for a payment process. System 100 can obtain authentication keys 118, CVC3 keys 403, and payment credentials 174 from a cloud-based transaction data generation system 106 and provisioning the retrieved payment credentials 174 to the MAA 111. The mobile device 104 can then complete transmission 308 to return the CVC3 values (Track 1 and Track 2) to the POS terminal 181 as part of a payment transaction for the selections 304 in shopping cart 306.

[0091] Payment at the POS terminal 181 then occurs using a standard process for payment transaction. For example the transaction acquisition processing by a transaction acquirer

166, payment processor 103, and issuer 180 can be carried out as described above with reference to Figure 1.

[0092] In the exemplary embodiments of Figures 6-8, an authorization request 168 can be routed from the POS terminal 181 to the acquirer 166, wherein the authorization request 168 includes DExx CVC3 track data 610 to facilitate payment between the POS terminal 181 and the acquirer 166 or a bank. In an embodiment, the track data DExx CVC3 610 can be DE35/DE45 (CVC3) track 2 or track 1 data. According to an embodiment Track 2 Data (DE 35) comprises information encoded on track 2 of a payment card's magnetic stripe as defined in ISO 7813, including field separators, but excludes beginning and ending sentinels and Longitudinal Redundancy Check (LRC) characters. In an embodiment, Track 1 Data (DE 45) includes information encoded on track 1 of a bankcard's magnetic stripe as defined in ISO 7813, including field separators. However, this excludes beginning and ending sentinels and LRC characters.

[0093] As discussed above with reference to Figure 1 and shown in Figures 3-8, the system 100 is configured to make use of the connection 178 between the issuer 180 and the payment credentials management system 114. As also shown in Figures 3-8, the issuer 180 in system 100 accesses the CVC3 keys 203.

[0094] The payment credentials management system 114 may manage the CVC3 keys 403. As shown in Figure 5, the authentication service 116 in system 100 may be a CAP Token Validation Service (CTVS). Upon completion of the authorization, the issuer 180 will respond back to the payment processor 103 (e.g., MasterCard) as described above with reference to Figure 1.

[0095] Figure 9A illustrates high level flow between components of the system 100 for completing a contactless payment from a mobile computing device without requiring an SE as part of a pull model. Figure 9A is described with continued reference to the embodiments illustrated in Figures 1, 2A and 3-8. However, Figure 9A is not limited to those embodiments.

[0096] The high level flow depicted in Figure 9A and described below assumes that the following operations have been completed: user registration; installation of a mobile payment

application; and an initialization process for the mobile payment application. As discussed above with reference to Figure 2A, this can be accomplished using the process 200. As shown in Figure 9A, once these operations have been completed, the pull environment is ready, and the next phase can take place using the following processes: a process to send authentication credentials and Keys 918 and any needed static payment credentials 928 to a user (cardholder 113) of the mobile device 104, a remote authentication 824, a process to retrieve dynamic payment credentials 938 (i.e., using an encrypted payment token payload 112); and a process to synchronize 978 the cloud-based transaction data generation system 106 and issuer 180 systems.

[0097] Finally, with continued reference to Figure 9A, a contactless payment transaction 908, such as a standard PayPass® Magstripe transaction, can take place with redemption of preloaded payment credentials and generation of a Dynamic Card Validation Code (CVC3). In Figure 9A, the contactless payment transaction 908 can use pre-computed credentials to perform a standard PayPass® Magstripe, which can use a generated dynamic card validation code (CVC3) (i.e., a Cryptogram).

[0098] As shown in Figure 9A, processing the contactless transaction 908 can include transaction acquisition 910 by a transaction acquirer 166 and authorization 912 by an issuer 180.

[0099] Exemplary solutions and embodiments disclosed herein can incorporate several core principles outlined below:

- Storage Key ($K_{Storage}$) defined at time of authentication profile and static Payment credentials provisioning
- Authentication credentials protected using MAA rules (e.g. Key Camouflage) (**Not** using $K_{Storage}$)

- SSL Layer between Mobile Payment Application and the cloud-based transaction data generation system 106 ("the Cloud")
    - Server Authentication using SSL
    - Client Authentication using CAP Token

Additional Storage Key used to counter Man-in-the-Middle attack (eavesdropping) at time of Payment Credentials provisioning from the Cloud to the Mobile Payment Application

- Authentication process between Mobile and Cloud to retrieve credentials
  - Identification (~ Virtual Card Profile ID (which may be any identifier defined by the Issuer and known by the cardholder 113 (e.g. Masked PAN…))
  - Authentication Transaction (e.g. Challenge / Response)
- CTVS validation
  - Successful > Use valid IMK_{CVC3} & IMK_{UN}
  - Failed > Use fake IMK_{CVC3} & IMK_{UN}

- The values KD_{CVC3} and $IVCVC3_{Track1/2}$ are static (if one considers a given PAN (and PSN) value, the values KDCVC3 and IVCVC3Track1/2 remain the same during the entire lifespan of the card. Those values are static. It also means that once the value is disclosed, you can reuse it.) for a given PAN (and PSN). The PSN (if available) can be part of the KDCVC3 derivation process. This avoids mandating any change regarding the management of this value at issuer level even if the PSN may be used to identify a SE-less 'virtual card' defined for a given PAN.
- The $CVC3_{Track1/2}$ is a dynamic data for a given PayPass® Transaction (UN, ATC, $IVCVC3_{Track1/2}$ and KD_{CVC3})
- Key derivation process in the Cloud (KD_{CVC3} + KD_{UN})
- Session key generation (KS_{UN}) in the Cloud to bind (ATC, UN_{CLOUD}, PAN and PSN) at the Edge (Mobile Payment Application)
- 'CVC3' generation in the Cloud (Using UN_{CLOUD}) >> $Cloud\_CVC3_{TRACK1/2}$
- Delivery of Encrypted Payload [using $K_{Storage}$] (KS_{UN}, $Cloud\_CVC3_{TRACK1/2}$ and ATC) to Mobile Payment Application

- CVC3 generation in the Mobile (Using UN_{READER} , KS_{UN}, Cloud_$CVC3_{TRACK1/2}$ and ATC)

Crypto

KD_{CVC3}

[00100] Concatenate from left to right the **PAN** (without any 'F' padding) with the **PSN** (if the PAN sequence number is not available, then it is replaced by a '00' byte). If the result X is less than 16 digits long, pad it to the left with hexadecimal zeros in order to obtain an eight-byte number Y in numeric (n) format. If X is at least 16 digits long, then Y consists of the 16 rightmost digits of X in numeric (n) format.

Generate KD_{CVC3} using:

$Z_L := DES3(IMK_{CVC3})[Y]$

$Z_R := DES3(IMK_{CVC3})[Y \oplus (\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'})]$

$KD_{CVC3} := (Z_L \| Z_R)$

This key is kept in the Cloud/Issuer (No disclosure to Mobile Payment Application)

KD_{UN}

Reuse Value Y as defined above

Generate KD_{UN} using:

$Z_L := DES3(IMK_{UN})[Y]$

$Z_R := DES3(IMK_{UN})[Y \oplus (\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'})]$

$KD_{UN} := (Z_L \| Z_R)$

This key is disclosed to the Mobile Payment Application (Some protection mechanisms can apply - e.g. Camouflage)

$IVCVC3_{TRACK1/2}$

$IVCVC3_{TRACK1}$ is a MAC calculated over the Track 1 Data using KD_{CVC3}

$IVCVC3_{TRACK2}$ is a MAC calculated over the Track 2 Data using KD_{CVC3}

Those values are kept in the Cloud/Issuer (No disclosure to Mobile Payment Application)

$Cloud\_CVC3_{TRACK1/2}$

1. Concatenate the following data to obtain an 8 byte data block (D):
   - $IVCVC3_{TRACK1/2}$ (2 bytes)
   - UN_{CLOUD} (4 bytes)
   - ATC (2 bytes)

2. Calculate O as follows:

   $O := DES3(KD_{CVC3})[D]$

   The two least significant bytes of O are the $CVC3_{TRACK1/2}$

   $CVC3_{TRACK1/2}$ generated in the Cloud are called $Cloud\_CVC3_{TRACK1/2}$

   UN_{CLOUD} is **not** sent to the Mobile device.

   UN_{CLOUD} is part of the payload exchanged between the Cloud and the Issuer.

   KS_{UN}

   Generate KS_{UN} using:

   $U_L := DES3(KD_{UN})[(ATC \| \text{'F0'} \| \text{'00'} \| UN_{CLOUD})]$

   $U_R := DES3(KD_{UN})[(ATC \| \text{'0F'} \| \text{'00'} \| UN_{CLOUD})]$

   $KS_{UN} := (U_L \| U_R)$

   (Static) Information known by the Mobile Payment Application

◊ FCI (PPSE)

◊ AID (Application Identifier)

◊ FCI (File Control Information)

◊ AFL (Application File Locator)

◊ AIP (Application Interchange Profile)

◊ AVN (Application Version Number)

◊ Encrypted (using $K_{Storage}$) Payment Credentials provisioned at time of authentication credentials provisioning. The Issuer should implement segregation rules in order to

prevent any use of leaked static payment credentials for CNP transactions (e.g. Misuse PAN for eCommerce / MOTO transactions).

- Track 1 Data
- Track 2 Data
- $PCVC3_{TRACK1/2}$
- $PUNATC_{TRACK1/2}$
- $NATC_{TRACK1/2}$

Encrypted (using $K_{Storage}$) Payload sent to the Mobile Payment Application (Valid for one contactless payment transaction)

◊ $Cloud\_CVC3_{TRACK1/2}$

◊ ATC

◊ KS_{UN}

Payload sent to the Issuer 180

◊ Identifier (PAN…)

◊ UN_{CLOUD}

◊ ATC

◊ Authentication Status Info + Additional Generation Information (e.g. Validity)

$CVC3_{TRACK1/2}$

Mobile Payment Application to perform CVC3 generation using:

◊ Information from the Reader

◊ Stored Information

◊ Credentials previously retrieved from the Cloud

CVC3 value to be included in Payment Authorization message (Track 2 (and Track 1) information).

UN_{READER} (4 bytes) >> Discard all but PUNATC-NATC least significant digits, padding to 8 digits with 0's.

1. Concatenate the following data to obtain an 8 byte data block (M):

- Cloud_CVC3$_{TRACK1/2}$ (2 bytes)
- UN$_{READER}$ (4 bytes)
- ATC (2 bytes)

2. Calculate T as follows:

   T := DES3(KS$_{UN}$)[M]

   The two least significant bytes of T are the CVC3$_{TRACK1/2}$

   Note:

◊ Cloud_CVC3$_{TRACK1/2}$ (2 bytes) is used instead of IVCVC3$_{TRACK1/2}$ (2 bytes)

◊ KS$_{UN}$ is used instead of KD$_{CVC3}$

◊ Binding between UN$_{READER}$ and UN$_{CLOUD}$ is implicitly done using the crypto (KS$_{UN}$)

Issuer Validation Process

[00101] An exemplary validation process is described below wherein the Issuer 180 uses the information provided in the payment transaction:

- Identifier – e.g. PAN Information
- UN$_{READER}$ (4 bytes) – Partial Information retrieved from Track data (Discretionary Information)
- ATC (2 bytes) – Partial Information retrieved from Track data (Discretionary Information)
- CVC3$_{TRACK1/2}$ – Partial Information retrieved from Track data (Discretionary Information)

[00102] The Identifier & ATC values are used to retrieve the information provided by the cloud-based transaction data generation system 106 ("the Cloud system"):

◊ Identifier (PAN,…)

◊ UN$_{CLOUD}$

◊ ATC

◊ Authentication Status Info + Additional Generation Information (e.g. Validity)

The Issuer system is able to compute **Cloud_CVC3$_{TRACK1/2}$** using:

◊ $IVCVC3_{TRACK1/2}$ (2 bytes)

◊ UN_{CLOUD} (4 bytes)

◊ ATC (2 bytes)

The Issuer system is able to compute $\mathbf{CVC3_{TRACK1/2}}$ using:

◊ $Cloud\_CVC3_{TRACK1/2}$ (2 bytes)

◊ UN_{READER} (4 bytes)

◊ ATC (2 bytes)

The Issuer can validate the $\mathbf{CVC3_{TRACK1/2}}$.

Push Model Flows and Communication Sequences

[00103] Figures 9B and 10-15 depict data flows and communication sequences between components of the system 100 using a push model. In an embodiment, the push model uses a standard notification mechanism to inform the consumer/cardholder 113 and allow retrieval of payment credentials managed in the cloud-based transaction data generation system 106 ("the Cloud").

[00104] In one embodiment, an access code (defined by cardholder 113) is used to load and use payment credentials, which can be stored/transported in an encrypted payment token payload 112.

[00105] As described below with reference to the exemplary embodiments depicted in Figures 9B and 10-15, implementations of the push model can include the following activities: registration and installation (described above with reference to Figure 2B and below with reference to Figure 10), initialization (described below with reference to Figure 11), a first (initial) provisioning (described below with reference to Figures 12A and 12B); subsequent provisionings after notification; and payment using a payment token payload that is valid for one transaction (also described below with reference to Figures 12A and 12B).

[0100] Figure 9B illustrates high level flow between components of the system 100 for completing a contactless payment from a mobile computing device without requiring an SE as part of a push model. Figure 9B is described with continued reference to the embodiments

illustrated in Figures 1, 2B and 3-8. However, Figure 9B is not limited to those embodiments.

[0101] The high level flow depicted in Figure 9B and described below assumes that the following operations have been completed: user registration; installation of a mobile payment application; and an initialization process for the mobile payment application. As shown in Figure 9B, once these operations have been completed, the environment is ready, and the next phase can take place using the following processes: a process to trigger provisioning 902, send a notification 904 to a user (cardholder 113) of the mobile device 104, a process to retrieve 906 the encrypted payment token payload 112; and a process to synchronize 978 the cloud-based transaction data generation system 106 and issuer 180 systems.

[0102] Finally, with continued reference to Figure 9B, a contactless payment transaction 908, such as a standard PayPass® Magstripe transaction, can take place with redemption of preloaded payment credentials and generation of a Dynamic Card Validation Code (CVC3). As shown in Figure 9B, processing the contactless transaction 908 can include transaction acquisition 910 by a transaction acquirer 166 and authorization 912 by an issuer 180.

[0103] Figure 10 illustrates a communications sequence for registering a cardholder 113, installing a mobile application, and providing an activation code for initializing the installed mobile application as part of a push model. Figure 10 is described with continued reference to the embodiments illustrated in Figures 1, 2B, 3-8 and 9B. However, Figure 10 is not limited to those embodiments.

[0104] In Figures 10, 11, 12A, 12B and 14, communications are depicted in stages between components and entities. The components and entities include an app store 1002, a browser 1004, a consumer 113, a mobile device 104, a mobile application 1011, a payment credentials management system 114 (part of a cloud-based transaction data generation system 106

[0105] The communications sequence shown in Figure 10 includes a protocol used to support a registration and installation activity for registering and installing a mobile application 1011 on a mobile device 104.

[0106] In an embodiment, the registration and installation activity is only performed one time. As shown in Figure 10, the process can be initiated using a browser 1004 (or a dedicated application provided by the issuer 180) running on a PC, tablet computer, gaming console, Internet-enabled television, or the mobile device 104.

[0107] As shown in stages [1] and [2] of Figure 10, the security level of the registration process can be improved using a Two-Factor authentication (2FA) mechanism to "sign" the registration request.

[0108] Embodiments can employ a wide range of authentication solutions using the concept of CAP Token (a One-Time Password/OTP) and an authentication mode.

[0107] As shown in Figure 10, a connection between the consumer 113 (using a browser 1004) and the cloud-based transaction data generation system 106 can be secured using SSL/TLS (Server Authentication).

[0108] The security level of the installation and validation process of the mobile application 1011 leverages standard solutions provided by the mobile vendors, such as, but not limited to Apple's App Store/iTunes site and Google's Android Market.

[0109] At the end of the communication sequence/process, the mobile application 1011 is ready for initialization and the consumer/user 113 (identified by a User ID) has received an activation code from the payment credentials management system 114 (which in turn is part of the cloud-based transaction data generation system 106).

[0110] In stage [1], the registration system is accessed. Stage [1] can include communications with the cloud-based transaction data generation system 106 and the issuer 180. In the exemplary embodiment of Figure 10, stage [1] uses a secure sockets layer (SSL)/Transport Layer Secure (TLS) connection. As part of stage [1], the SSL/TLS connection is validated (i.e., by using certificates) and if validation is successful, a supplied user ID for the cardholder 113 and a bank account are used to select a product (e.g., the PayPass® PrePaid product in the exemplary embodiment of Figure 10). In an embodiment, an optional step uses Two-Factor authentication (2FA) to « sign » the request (according to standard eBanking and mBanking procedures) and generate a one-time password (OTP).

[0111] With continued reference to Figure 10, stage [2] encompasses consumer registration. Stage [2] can include communications with the cloud-based transaction data generation system 106 and the issuer 180. The consumer registration of stage [2] is performed by validating the request. As an optional step, stage [2] can comprise validating the OTP (using 2FA). This embodiment may require a connection with the issuer 180 system.

[0112] In stage [3], if the consumer registration is completed, a consumer profile is created in the cloud-based transaction data generation system 106 and the issuer 180 systems.

[0113] In stage [4], if the a consumer profile is created successfully, the consumer registration is completed (using an SSL/TLS Connection in the exemplary embodiment of Figure 10) and a unique activation code associated with the consumer profile (User ID) is returned. In an embodiment, this activation code can only be used one time for one activation.

[0114] In stage [5], an activation code is received (i.e., via the SSL/TLS Connection and the activation code is 'remembered.'

[0115] In stage [6], the mobile application 1011 is downloaded from an App Store 1002 and in stage [7] the mobile application 1011 is validated and installed.

[0116] At this point, in stage [8], the mobile application 1011 is ready to be initialized.

[0117] Figure 11 illustrates a communications sequence for activating an account and service, defining an access code, and initializing a mobile application as part of a push model. Figure 11 is described with continued reference to the embodiments illustrated in Figures 1, 2B and 3-10. However, Figure 11 is not limited to those embodiments.

[0118] Figure 11 describes a protocol used to support the initialization activity.

[0119] According to an embodiment, this activity is only performed one time.

[0120] This initialization process must be initiated using the Mobile Payment Application (mobile application 1011) installed on the mobile device 104 of the consumer 113 (i.e., user).

[0121] As shown in stages [2] and [5] of Figure 11, the security level of the initialization process can be improved using a Two-Factor authentication (2FA) mechanism to “sign” the initialization request. In an embodiment, an integrity check and rooted device check can be added to stage [2].

[0122] As illustrated in Figure 11, the connection between the mobile application 1011 and the cloud-based transaction data generation system 106 can be secured using SSL/TLS (Server Authentication).

[0123] The URL used to establish the connection can be defined in the mobile application 1011.

[0124] In the exemplary embodiment provided in Figure 11, the registration to the remote notification service 915 requires a unique mobile ID. Standard functions (i.e., application programming interface/API calls) can be used to generate this value.

[0125] In an embodiment, the consumer 113 has to define an access code that will be used to grant access to the payment token payload (which is retrieved and stored in encrypted form).

[0126] As shown in Figure 11, several parameters are pushed to the mobile application 1011 during the initialization process.

[0127] The communication sequence begins in stage [1] when the mobile application 1011 is started (the first use of the application).

[0128] In stage [2], consumer credentials are asked for (i.e., via prompts in GUI 202). According to the exemplary embodiment of Figure 11, the consumer credentials can include: a user ID, an activation code (provided during the registration process described above with reference to Figure 10), an access code (provided as part of a user interface/UI process to enter a value twice over and check the entry). Stage [2] can optionally use 2FA to « sign » the request (standard e/mBanking) and generate an OTP.

[0129] In stage [3], a unique mobile ID is generated. This ID can be used for remote notification.

[0130] Next, in stage [4], a connection to the cloud-based transaction data generation system 106 using a default URL defined in the mobile application 1011 is established via an SSL/TLS Connection. The SSL/TLS connection is validated (for example, by using certificates), and if the validation is successful, initialization credentials are sent. These credentials can include the user ID and an activation code (in an embodiment, this can only be used one time), an access code, and the mobile ID. When 2FA is used, an optional

initialization credential can include a provided OTP. At this point, the access code value is wiped from memory.

[0131] In stage [5], the activation code is validated for the User ID. In the exemplary embodiment of Figure 11, the activation code can only be used one time for one activation. The optional OTP from stage [4] is also validated (if used). If the validation is successful, the access code (used to compute the MA_Key) and the mobile ID (used for notification) are both stored.

[0132] In stage [6], the mobile ID is registered to the notification service before proceeding to stage [7].

[0133] In stage [7], information is generated and added to the consumer profile.

[0134] Application ID (Unique ID to access Consumer Profile)

[0135] The communications sequence depicted in Figure 11 encompasses pushing the following parameters/data to the mobile application 1011 as part of stage [7]: an application ID, which is a unique ID used to access the consumer profile (i.e., the profile for the user 113); a salt, which is a value used (in combination with the access code) in the cryptographic process (Fn_MA_Key) to generate the key used for transport and storage of the payment token payload; payment parameters including the required payment card artwork with a masked PAN value (e.g., XXXX XXXX XXXX 4321); a notification URL used to connect to the cloud-based transaction data generation system 106 to retrieve the encrypted payment token payload; and a card ID, which is a unique ID used in the generation (Fn_Auth_Code) of an authentication code. In an embodiment, the parameters optionally further comprise some additional non-sensitive static payment credentials.

[0136] In stage [8], this information is returned to the mobile application 1011 via an SSL/TLS connection before proceeding to stage [9] where the information is stored in the mobile application 1011.

[0137] As shown in Figure 11, the information stored in stage [9], can include: an application ID, the salt, payment parameters (e.g. the Masked PAN), the notification URL, and the card ID. Optional information stored can include static payment credentials (e.g. FCI (PPSE), AID shown in Figure 11).

[0138] At the end of the communications sequence of Figure 11 in stage [10], the mobile application 1011 is ready to receive a notification to start a first provisioning of payment credentials. In stage [11], rules to trigger this provisioning process are communicated. This first provisioning is described below with reference to Figures 12A, 12B and 13.

[0139] Figures 12A and 12B illustrate a communications sequence of a (first) provisioning of payment credentials as part of a push model, in accordance with an exemplary embodiment of the present disclosure. Figures 12A and 12B are described with continued reference to the embodiments illustrated in Figures 1, 2B and 3-12. However, Figures 12A and 12B are not limited to those embodiments.

[0140] Figures 12A and 12B depict the communications sequence to complete a (first) provisioning and subsequent provisionings after notification.

[0141] Figures 12A and 12B illustrate the protocol used to support the provisioning activity.

[0142] In an embodiment, this process can only be initiated after a notification is sent to the registered mobile application 1011.

[0143] As shown in Figure 12A, some default values (proof and ATC) are used for the first provisioning.

[0144] In embodiments, the provisioning activity depicted in stages [1]-[22] in Figures 12A and 12B and reference numerals 1-22 of Figure 13 covers the following:

- Delivery of an encrypted Session ID
- The connection between the mobile application 1011 and the cloud-based transaction data generation system 106 ("the Cloud") is secured using SSL/TLS (Server Authentication).

  The URL used to establish the connection to the Cloud was provided at time of the Initialization (it must not be part of the notification message).
- Generation of an ***Authentication Code*** that demonstrates (using a One-Way function):

- ◊ The ability to generate the key (MA_Key) using a Salt (stored as a parameter in the Mobile Payment Application) and the Access Code to be provided by the Consumer (= User)
- ◊ The ability to decrypt the content of the Notification Message to extract the ***Session ID***
- ◊ The knowledge of the last ***Proof*** value provided as part of the last received encrypted Payment Token Payload
- ◊ The knowledge of parameters provided at time of the Initialization process (***Card ID***)

♦ Retrieval of ***Encrypted Payment Token Payload*** after successful validation of the Authentication Code and generation of the Encrypted Payment Token Payload (

*Fn_**Proof**, Fn_IVS_CVC3* and *Fn_KS_CVC3*)

- Generation of an ***Activation Proof*** that demonstrate the delivery of the Payment Credentials to the Mobile Payment Application
- Activation of the Payment Credentials on the Issuer Validation systems after successful validation of the Activation Proof. This process is part of the ***synchronization*** between the Cloud and the Issuer systems
- Activation of the Payment Credentials (set to enabled in the mobile payment Application) at the end of the process

[0145] The provisioning activity illustrated in Figures 12A and 12B is defined to mitigate the risk of Mobile Cloning.

[0146] It is also designed to support the retrieval of more than one payment token payload. Issuer rules can be defined to manage and limit the generation of payment credentials.

[0147] The provisioning also covers the detection of any misuse of Payment Credentials (e.g. duplicate/replay detection, loss of sequence when checking ATC…).

[0148] The generated encrypted payment token payload 112 contains sensitive data including Track Data, which in turn contains a PAN value.

[0149] It is inherent to the PayPass® Magstripe Transaction 908 process that the encrypted payment token payload 112 will have to be decrypted to retrieve those sensitive data which are sent in 'clear' to a PayPass® Reader the same way it is done when using a PayPass® Magstripe card.

[0150] Stages [1]-[22] of the exemplary embodiments illustrated in Figures 12A and 12B are described below. As further noted below with reference to Figure 13, stages [1]-[22] correlate to reference numerals 1-22 shown in the flow diagram for retrieving an encrypted payment token payload 112 after notification.

[0151] Stage [1] is the trigger to start the provisioning process before proceeding to stage [2] where data is prepared for remote notification. The data prepared in stage [2] can include generation and storage of a session ID (7 bytes in the exemplary embodiment of Figure 12A)

for an application ID. An optional data item is the definition of validity (in conjunction with rules in the cloud-based transaction data generation system 106) for a Session ID

[0152] In stage [2], a MA_Key is computed using Fn_MA_Key. In this stage, the access code and salt are also prepared. The message is formatted (based upon the length + session ID) (8 bytes in the example of Figure 12A) and the message is then encrypted using the MA_Key. The notification message is built using E(Msg).

[0153] In stage [3], a notification is sent to the mobile ID before proceeding to stage [4].

[0154] In stage [4], the notification is received with E(Msg). The user interface/UI (at the operating system/OS level) can then display the notification alert.

[0155] In stage [5], the mobile application 1011 is started. This can entail using GUI 202 to inform the user 113 along with a UI element to request or prompt the user 113 for the access code.

[0156] In stage [6], the requested access code is provided and the communication sequence for provisioning proceeds to stage [7].

[0157] In stage [7], process notification occurs. As shown in Figure 12A, this can include computing the MA_Key using Fn_MA_Key (access code, salt), wiping the access code from memory, decrypting the E(Msg) using the MA_Key, validating the length, and extracting the Session ID.

[0158] In stage [8], a connection to the cloud-based transaction data generation system 106 is established using URL Notification (can be over a SSL/TLS Connection) and the SSL/TLS connection is validated (can be accomplished using certificates).

[0159] In stage [9], the authentication code is computed using Fn_Auth_Code. As shown in Figure 12A, this can involve use of the card ID, the session ID, and the last known proof (default value is 'FFFFFFFFFF') associated with a last known ATC (default value = '0000'). At this point the following authentication credentials are sent: the application ID, the authentication code, and the last known ATC (Default Value = '0000') before the authentication code is wiped from memory.

[0160] In stage [10], the authentication credentials are validated by checking rules for the ATC to accept or reject the payment token payload generation. If the payment token payload

generation is accepted, the authentication code is computed using Fn_Auth_Code. As shown in Figure 12A, this can entail using the card ID, the session ID, and the proof associated with the received ATC. At this point the authentication code is checked.

[0161] In stage [11], the payment token payload is built. In the exemplary embodiment of Figure 12A this stage includes retrieving payment static information from the consumer profile and generating dynamic data. Such dynamic data generation includes: the proof (using Fn_Proof), the IVS_CVC3 (using Fn_IVS_CVC3), incrementing the ATC, and the KS_CVC3 (Using Fn_KS_CVC3).

[0162] At this point the ATC and proof are stored as the last known ATC and last known proof, respectively.

[0163] As illustrated in Figure 12A, the payment token payload contains Track Data, which in turn contains a PAN value (as well as some other credentials).

[0164] In the embodiment of Figure 12A, the payment token payload is always exchanged encrypted between the cloud-based transaction data generation system 106 and the mobile application 1011 and the mobile application 1011 always stores this value encrypted.

[0165] The standard PayPass® Magstripe Transaction requires that the Track data is sent in clear between the mobile application 1011 and the PayPass® Reader. This is also true for any PayPass® Card. In this way, the protocol illustrated in the communication sequence of Figures 12A and 12B aims reducing the exposure of sensitive data to a very limited time (i.e., a small 'window' of time).

[0166] Stage [11] continues with the computation of the MA_Key using Fn_MA_Key (Access Code, Salt), formatting of the message Msg (Length + Payment Token Payload) and encryption of Msg using MA_Key before proceeding to stage [12] where the E(Msg) is returned to the mobile application 1011 over the SSL/TLS Connection.

[0167] In stage [13] the E(Msg) is validated and then decrypted using MA_Key. In this stage the Length is validated the [End Tag] is extracted and validated. Next, Extract Proof and ATC before wiping the Decrypted Msg and if this is OK, stage [13] stores the ATC + E(Msg) (the encrypted payment token payload 112) with a status of 'Not yet enabled.' At

this point, the ATC and the Proof are stored as the Last known ATC and Last known Proof, respectively.

[0168] Next, the provisioning process performs the following communications depicted as stages [13]-[22] in Figure 12B.

[0169] Stage [13] includes the following: Validate E(Msg) and Decrypt E(Msg) using MA_Key by: validating Length, extracting and validating [End Tag], extracting Proof and ATC. Next, the decrypted Msg is wiped. Once this is done, the ATC + E(Msg) are stored (= Encrypted Payment Token Payload) with a status of 'Not yet enabled.' The ATC is stored as Last known ATC and the Proof is stored as Last known Proof.

[0170] In stage [14], the activation proof is generated by formatting the Msg (Length + Proof + ATC) and encrypting the Msg using the MA_Key.

[0171] In stage [15], the Activation Proof is sent via the SSL/TLS connection before proceeding to stage [16].

[0172] In stage [16], the activation proof is validated by decrypting the E(Msg) using MA_Key in order to validate the Length and extract and check the proof and the ATC against the Last known Proof and Last known ATC.

[0173] In stage [17], the payment token payload is activated by setting 'Enabled' in the cloud-based transaction data generation system 106. Then, the issuer 180 is provided (informed) with information (ATC, UNCloud, ...) needed to compute the IVS_CVC3 and KS_CVC3. In stage [18], this information is stored for a subsequent validation process.

[0174] In stage [19], the mobile application 1011 is informed (via a return code sent over the SSL/TLS connection.

[0175] In stage [20], the return code is analyzed and if it is OK, the payment token payload status is set to 'Enabled.'

[0176] In stage [21], an optional step continues to load the payment token payload as needed by returning control to stage [9]. Alternatively, in stage [21], the MA_Key is wiped from memory before continuing with stage [22].

[0177] In stage [22], the mobile application 1011 is ready to enable payment. This can include informing the cardholder 113 via a UI element in the GUI 202.

[0178] The protocol shown in Figures 12A and 12B and in the payment activity illustrated in Figures 14 and 15 is designed to reduce the time when those assets are stored in the memory of the mobile device 104 and used by the mobile payment application.

[0179] Figure 13 is a diagram of a system illustrating flows between system components to retrieve an encrypted payment token payload after notification as part of a (first) provisioning of payment credentials for a push model, in accordance with an exemplary embodiment of the present disclosure. Figure 13 is described with continued reference to the embodiments illustrated in Figures 1, 2B, 3-11, 12A and 12B. However, Figure 13 is not limited to those embodiments.

[0180] In Figure 13, reference numbers 1-22 in the flow correlate to stages [1]-[22] of the provisioning activity and communications sequence depicted in Figures 12A and 12B described above. The values A_1, A_2 and B in Figure 13 denote information available before the start of the provisioning activity.

[0181] At the end of the flow depicted in Figure 13 (i.e., after steps 1-22 have been completed), the mobile application 1011 is ready to perform a payment or to receive subsequent notifications.

[0182] Figure 14 illustrates a communications sequence for using a mobile payment application to access a locally stored encrypted payment token payload to process a standard PayPass® Magstripe transaction as part of a push model, in accordance with an exemplary embodiment of the present disclosure. Figure 15 is described with continued reference to the embodiments illustrated in Figures 1, 2B, 9B-12, 12A, 12B and 13. However, Figure 14 is not limited to those embodiments.

[0183] Figure 14 illustrates the communications sequence and protocol used to support the Payment activity. This process is initiated by the consumer 113 (i.e., the user) using a mobile application 1011 properly initialized and provisioned with encrypted payment token payload. The Encrypted Payment Token Payload is **NOT** sent to a PayPass® Reader. The payment token payload is a "token" that can be used to perform one payment transaction. A defined process uses the content of this payload to support a standard PayPass® Magstripe transaction 908.

[0184] As shown in the exemplary embodiment of Figure 14, the Payment activity covers the following:

- The start of the Mobile Payment Application with Integrity and Rooted Device checks. In an embodiment, this can be done by adding a test/check for the Rooted Device in the initialization phase.
- The Entry of the ***Access Code*** to grant access to the content of the encrypted Payment Token Payload
- Enabling of the ***NFC*** interface of the Mobile Device
- Processing a standard PayPass® Magstripe Transaction 908 using the Mobile Payment Application with the delivery of Payment Credentials and a Dynamic Card Validation Code (CVC3) (*Fn_GenCVC3*).
- Online PIN entry using a PED (PIN entry Device) at the POS when the online PIN is used
- Standard authorization process with the actors or the Payment Ecosystem (Merchant 181 – Acquirer 166 – Payment processing network 170 – Issuer 180)
- CVC3 Validation process (*Fn_ValCVC3*) by the Issuer using information provided by the Cloud at time of the synchronization process; and
- An opportunity to use the Remote Notification process to push information back to the consumer 113 (i.e., the user).

[0185] Stages [1]-[17] of the payment communications sequence are described below in accordance with the exemplary embodiment illustrated in Figure 14.

[0186] Stage [1] the mobile application 1011 is started.

[0187] In stage [2], the mobile application 1011 performs an integrity Check / rooted device check, uses the GUI 202 to Inform User 113, and checks to see if the Payment Token Payload is available. If so, a UI element in the GUI 202 requests the Access Code.

[0188] In stage [3], the access code is provided and in stage [4] the Encrypted Payment Token Payload is loaded by retrieving the Encrypted Payment Token Payload associated with the lowest ATC Value. As noted in Figure 14, and described below with reference to stage

[8], stage [4] can optionally include the MA_Key Computation and Payment Token Payload Decryption as needed for performance optimization.

[0189] In stage [5], the NFC Interface is enabled and a UI element of the GUI 202 is used to inform the User 113 (the application is Ready for Payment). In stage [6] an NFC tap is detected and the process proceeds to stage [7].

[0190] In stage [7], a PayPass® Magstripe Transaction 908 (using a PayPass® Reader) is initiated using UN_{Reader}.

[0191] In stage [8] PayPass® Magstripe Transaction 908 is performed using the mobile application 1011.

[0192] In this stage, the MA_Key is computed using Fn_MA_Key (Access Code, Salt). The Encrypted Payment Token Payload is decrypted, the MA_Key is wiped from memory, and the Length is validated.

[0193] Next, provided the validation is successful, Payment Credentials are extracted.

[0194] Stage [8] supports the following functions:

- PPSE Management
- SELECT AID
- GET PROCESSING OPTIONS
- READ RECORD
- COMPUTE CRYPTOGRAPHIC CHECKSUM

[0195] Next, the CVC3 is generated using Fn_GenCVC3 and data is returned to PayPass Reader (PPSE, SELECT, GPO, READ RECORD, CCC) before wiping the Decrypted Payment Token Payload. As shown in Figure 14, a UI element of the GUI 202 can be used to inform the user 113 of the status after the transaction.

[0196] As noted in Figure 14 with reference to stage [8], in an embodiment, the MA_Key Computation and Payment Token Payload Decryption may be moved to Stage [4] in case of performance issues.

[0197] If these operations are moved to stage [4], a timer can be defined to limit the potential exposure of sensitive data. For example, there can be a constraint set to complete a

PayPass® Magstripe Transaction in less than 250 milliseconds (ms) [170 ms allowed for the Card (in this case, the mobile application 1011) and 80 ms allowed for the PayPass® Reader].

[0198] In stage [9], an optional communication provides the Online PIN.

[0199] In stage [10], a standard process for a PayPass® Magstripe Transaction 908 is performed between the merchant 181 and the transaction acquirer 166 (see reference number 10 of the flow diagram of Figure 15).

[0200] In stage [11], the standard process for the PayPass® Magstripe Transaction 908 is carried out between the acquirer 166 and the payment processing network 170 (see reference 11 of the flow diagram of Figure 15).

[0201] In stage [12], the standard process for the PayPass® Magstripe Transaction 908 is carried out between the payment processing network 170 and the Issuer 180 (see number 12 in Figure 15).

[0202] At this point, in stage [13], the Issuer 180 is ready to validate the PayPass® Magstripe Transaction 908 (by completing the Technical Authorization).

[0203] Next, in stage [14], CVC3 validation is done using Fn_ValCVC3 before proceeding to stage [15], where the standard process for PayPass Magstripe Transaction is completed between the issuer 180, the payment processing network 170, the Acquirer 166, and the Merchant 181 (see reference number 14 in Figure 15).

[0204] In stage [16], the cloud-based transaction data generation system 106 is Informed (e.g., using a Trigger Push of new Payment Token Payload or Information/Alert to the consumer 113).

[0205] At this point, in stage [17], the mobile application 1011 is ready for a new payment (if Payment Token Payload available) or for a new notification (via a Push Payment Token Payload).

[0206] Figure 15 is a diagram of system 100 illustrating flows between system components to use a mobile payment application to access a locally stored encrypted payment token payload to process a standard PayPass® Magstripe payment transaction as part of a push model. Figure 15 is described with continued reference to the embodiments illustrated in

Figures 1, 2B, 9-12, 12A, 12B, 13 and 14. However, Figure 15 is not limited to those embodiments.

[0207] Figure 15 provides a flow view within the system 100 of the payment protocol described above with reference to Figure 14. In Figure 15, the reference numerals 1-17 are used to describe the stages [1]-[17] of the Payment activity depicted in Figure 14.

[0208] In Figure 15, the values A1, A2 and B refer to the information available before the start of the payment activity.

[0209] At the end of the flow process depicted in Figure 15, the Mobile Application is ready to perform another Payment (if at least one encrypted Payment Token Payload is available) or to receive subsequent notification(s) to retrieve encrypted Payment Token Payload from the cloud-based transaction data generation system 106.

[0210] The timing constraints to perform a PayPass® Magstripe transaction 908 may require that some processing (i.e., MA_Key generation and decryption of the encrypted Payment Token Payload) is performed before the NFC Tap. In this case, a timer can be defined and monitored in order to control the time when sensitive data might be exposed in the memory of the mobile device 104.

[0211] The following section provides further details for exemplary embodiments of the parameters, data elements and fields depicted in Figures 9-15 and described above with reference to those Figures.

[0212] Payment Token Payload

According to an embodiment, the Payment Token Payload contains four parts:

- Length
- Proof Information
- Payment Data (Static – Non Sensitive, Static – Sensitive, Dynamic)
- [End Tag]

The Payment Token Payload is transported and stored in encrypted form using a Mobile Application Key (MA_Key). The process to derive this key is defined in *Fn_MA_Key*.

The ***Length*** field is used for control during the decryption process.

The same remark applies for the ***[End Tag]***. This value is used to identify the 'end' (= Tail) of the Payment Token Payload.

The ***Proof Information*** contains information used to:

- Grant access to the provisioning process (using the concept of Authentication Credentials as defined in *Fn_Auth_Code*)
- Supply an activation proof to the cloud-based transaction data generation system 106 (""the Cloud") after retrieval of an encrypted Payment Token Payload
- Use random information in the first block of data to be encrypted using a Cipher Block Chaining (CBC) mode encryption

[0213] The ***Payment Data*** contains all the data elements required to perform a *PayPass* Magstripe Transaction. In an embodiment, the detailed content of the Payment Token Payload is:

- Length
- *Proof Information*
  - Proof (Random – 5 bytes)
- *Static Payment Data*
  - FCI (PPSE)
  - AID
  - FCI (*PayPass* App)
  - AIP
  - AFL
- *Static Payment Data (PayPass Transaction)*
  - PUNATC Track 1
  - PUNATC Track 2
  - PCVC3 Track 1
  - PCVC3 Track 2
  - NATC Track 1
  - NATC Track 2
  - UDOL
- *Static Payment Data (Sensitive Data)*
  - Track 1 Data
  - Track 2 Data
- *Dynamic Payment Data (Sensitive Data)*
  - IVS_CVC3 (Track 1 and Track 2)
  - ATC
  - KS_CVC3

- *Control Data*
  - [End Tag]

Core Functions

[0214] What follows is a description of exemplary embodiments of core functions used by the push model described above with reference to Figures 2B and 9-15.

Overview of Functions and their use

Fn_MA_Key Generate MA_Key (used for encryption during Transport and Storage)

Fn_Auth_Code Generate Authentication Code value (used to authenticate Mobile Application when replying to Remote Notification)

*Fn*_**Proof** Generate Proof value (used to validate and enable Payment Token Payload)

Fn_IVS_CVC3 Generate dynamic IV_CVC3 value (Valid for one transaction)

Fn_KS_CVC3 Generate dynamic KS_CVC3 key (Valid for one transaction)

Fn_GenCVC3 Generate CVC3 using UN_{Reader} and Payment Token Payload

Fn_ValCVC3 Validate CVC3 generated using UN_{Reader} and Payment Token Payload

Fn_MA_Key

This function is used to generate MA_Key.

The MA_Key key is used for the protection of the Payment Token Payload during Transport and Storage.

The basic concept is using a ***Password-Based Key Derivation Function*** to generate a key using:

- ***Salt***
  The Salt is defined by the Cloud and sent to the Mobile Application at time of the initialization process.
  It must be at least 64 bits
- ***Access Code***
  The Access Code is defined by the Consumer (= User) at time of the Initialization process.
  It must be at least 6 characters.

The combination of the Access Code and the Salt must provide a sufficient security level compared to the expected User Convenience.

The inputs for Key Derivation are:

- Access Code
- Salt
- Parameters (e.g. Number of iterations, key length…)

The following functions may be used to support the Password-Based Key Derivation Function algorithms:

- PBKDF2
- *bcrypt*

- *scrypt*

A typical use is:

MA_Key := PBKDF2(PseudoRandomFunction, Access Code, Salt, Parameters)

As stated above, this key is primarily used to protect the Payment Token Payload.

It is also used to protection the delivery of the Session ID in the Notification message sent to the Mobile Device (and the Mobile Payment Application)

According to embodiments, the push model described herein can use AES or DES3 as symmetric-key algorithms. The length of MA_Key will be set accordingly to the chosen algorithm (i.e., AES or DES3).

Fn_Auth_Code

[0215] As described above with reference to the (First) Provisioning and Subsequent Provisions after Notification flow and communication sequence of Figures 12A, 12B and 13, the generation of an ***Authentication Code*** value can be used to authenticate the Mobile Payment Application when replying to a Remote Notification.

[0216] In an embodiment, the Authentication Code is the result of a SHA-256 function using the following parameters:

- Card ID
- Session ID
- Last known Proof

Fn_Proof

[0217] The ***Proof*** is a random value (5 bytes) generated by the cloud-based transaction data generation system 106.

[0218] The Proof is used when generating an ***Authentication Code*** value.

[0219] The Authentication Code is used to authenticate the Consumer (= User) and the Mobile Application at time of retrieval of new Payment Token Payload.

[0220] The Proof is also used to generate an ***Activation Proof*** value.

[0221] The Activation Proof is used to enable a received Payment Token Payload.

Fn_IVS_CVC3

[0222] According to an embodiment, the push model described herein can use Dynamic IV_CVC3 values (Track 1 and Track 2) in order to prevent any disclose of the IV_CVC3 values to an environment without requiring use of a Secure Element.

[0223] These Dynamic IV_CVC3 values (**IVS_CVC3**) are valid for one transaction.

[0224] A first CVC3 generation (using UN_{Cloud}) in the Cloud is used to support the generation of the IVS_CVC3 values.

KD_{CVC3}

Concatenate from left to right the **PAN** (without any 'F' padding) with the **PSN** (if the PAN sequence number is not available, then it is replaced by a '00' byte).

If the result X is less than 16 digits long, pad it to the left with hexadecimal zeros in order to obtain an eight-byte number Y in numeric (n) format.

If X is at least 16 digits long, then Y consists of the 16 rightmost digits of X in numeric (n) format.

Generate KD_{CVC3} using:

$Z_L := DES3(IMK_{CVC3})[Y]$

$Z_R := DES3(IMK_{CVC3})[Y \oplus (\text{'FF'} \| \text{'FF'} \| \text{'FF'} \| \text{'FF'} \| \text{'FF'} \| \text{'FF'} \| \text{'FF'} \| \text{'FF'})]$

$KD_{CVC3} := (Z_L \| Z_R)$

This key is kept in the Cloud/Issuer (No disclosure to Mobile Payment Application)

$IVCVC3_{TRACK1/2}$

$IVCVC3_{TRACK1}$ is a MAC calculated over the Track 1 Data using KD_{CVC3}

$IVCVC3_{TRACK2}$ is a MAC calculated over the Track 2 Data using KD_{CVC3}

Those values are kept in the Cloud/Issuer (No disclosure to Mobile Payment Application)

$Cloud\_CVC3_{TRACK1/2}$

3. Concatenate the following data to obtain an 8 byte data block (D):
   - $IVCVC3_{TRACK1/2}$ (2 bytes)
   - UN_{CLOUD} (4 bytes)
   - ATC (2 bytes)
4. Calculate O as follows:

   $O := DES3(KD_{CVC3})[D]$

   The two least significant bytes of O are the $CVC3_{TRACK1/2}$

   IVS_CVC3 := $CVC3_{TRACK1/2}$ generated in the Cloud

   *UN_{CLOUD} is **not** sent to the Mobile Payment Application.*

   UN_{CLOUD} is part of the payload exchanged between the Cloud and the Issuer during the Synchronization process.

Fn_KS_CVC3

[0225] In an embodiment, the push model described herein can use the "Session" KD_CVC3 key in order to prevent any disclose of the KD_CVC3 key to an environment without requiring use of a Secure Element.

[0226] This "Session" KD_CVC3 key (**KS_CVC3**) is valid for one transaction.

A key derivation (using UN_{Cloud}) in the Cloud is used to support the generation of the KS_CVC3 key.

KD_{UN}

Concatenate from left to right the **PAN** (without any 'F' padding) with the **PSN** (if the PAN sequence number is not available, then it is replaced by a '00' byte).

If the result X is less than 16 digits long, pad it to the left with hexadecimal zeros in order to obtain an eight-byte number Y in numeric (n) format.

If X is at least 16 digits long, then Y consists of the 16 rightmost digits of X in numeric (n) format.

Generate KD_{UN} using:

$Z_L := DES3(IMK_{UN})[Y]$

$Z_R := DES3(IMK_{UN})[Y \oplus (\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'}\|\text{'FF'})]$

$KD_{UN} := (Z_L \| Z_R)$

KS_{UN}

Generate KS_{UN} using:

$U_L := DES3(KD_{UN})[(ATC \| \text{'F0'} \| \text{'00'} \| UN_{CLOUD})]$

$U_R := DES3(KD_{UN})[(ATC \| \text{'0F'} \| \text{'00'} \| UN_{CLOUD})]$

$KS_{UN} := (U_L \| U_R)$

KS_CVC3 := KS_{UN} generated in the Cloud

<u>Fn_GenCVC3</u>

This function generates CVC3 values (Track 1 and Track 2) using UN_{Reader} and the content of the Payment Token Payload.

The CVC3 values are included in Payment Authorization message (Track 2 (and Track 1) information).

The UN_{READER} (4 bytes) is also included using a process where we discard all but PUNATC-NATC least significant digits with a padding to 8 digits with 0's.

3. Concatenate the following data to obtain an 8 byte data block (M):
   - **IVS_CVC3** (2 bytes)
   - UN_{READER} (4 bytes)
   - ATC (2 bytes)
4. Calculate T as follows:

   T := DES3(**KS_CVC3**)[M]

The two least significant bytes of T are the $CVC3_{TRACK1/2}$

Note:

◊ **IVS_CVC3** (2 bytes) is used instead of $IVCVC3_{TRACK1/2}$ (2 bytes)
◊ **KS_CVC3** is used instead of KD_{CVC3}
◊ The binding between UN_{READER} and UN_{CLOUD} is implicitly done using the crypto (**KS_CVC3**)

Fn_ValCVC3

This function validates CVC3 values (Track 1 and Track 2) generated by the Mobile Payment Application using UN_{Reader} and the content of the Payment Token Payload.

For the validation process, the Issuer uses the information provided in the payment transaction:

- Identifier – e.g. PAN Information retrieved from Track Data
- UN_{READER} (4 bytes) – Partial Information retrieved from Track data (Discretionary Information)
- ATC (2 bytes) – Partial Information retrieved from Track data (Discretionary Information)
- $CVC3_{TRACK1/2}$ – Partial Information retrieved from Track data (Discretionary Information)

The Identifier and ATC values are used to retrieve the information provided by the cloud-based transaction data generation system 106 ("the Cloud system"):

◊ Identifier (PAN,…)
◊ UN_{CLOUD}
◊ ATC
◊ (Option) Additional Generation Information

The issuer 180 system is able to compute **IVS_CVC3** using:

◊ $IVCVC3_{TRACK1/2}$ (2 bytes)
◊ UN_{CLOUD} (4 bytes)
◊ ATC (2 bytes)
◊ KD_{CVC3}

The Issuer system is able to compute $\mathbf{CVC3_{TRACK1/2}}$ using:

◊ **IVS_CVC3** (2 bytes)
◊ UN_{READER} (4 bytes)

◊ ATC (2 bytes)
◊ **KS_CVC3**

The issuer 180 can validate the $\mathbf{CVC3_{TRACK1/2}}$ value.

[0227] An exemplary Notification Process is described below. The Remote Notification is a standard process used to ***push*** information to a mobile device 104 (and a mobile application 1011).

Notification Payload

[0228] The notification Payload cannot typically exceed 256 bytes and cannot be used to carry sensitive data.

[0229] According to an embodiment, the notification can be used to inform the Consumer 113 (= User) that data are available in the cloud-based transaction data generation system 106 ("the Cloud") and can be generated by the remote notification service 915.

[0230] The notification message can be built using a JSON dictionary object (RFC 4627). The code Example below illustrates the code for loading 3 payloads using a message to be processed using localization options:

```
        {
    "aps":{
        "alert":{
            "loc-key":"NEW_TOKEN"
        },
        "badge":3
    },
    "acme":"<<CODE>>"
}
```

Where <<CODE>> is the encrypted Session ID as defined in the (First) Provisioning and Subsequent Provisions after Notification described above with reference to Figures 12A, 12B and 13.

[0231] Remote notification solutions available for the following development platforms:

- J2ME
- Android
- iOS
- Windows Phone
- BlackBerry

Alternative Embodiment Processing and Communication Flows

[0232] FIG. 17 illustrates an alternative embodiment for processing mobile payments without the use of secure elements in a mobile device.

[0233] The cardholder 113 may utilize the mobile device 104, loaded with the mobile application 1011 and having no secure element payment application, to engage in a financial transaction via contactless payment. In such an embodiment, the payment token payload (PTP) 112 may be a container used to carry payment credentials, which may include a card profile 1702 and a single use key 1704.

[0234] The card profile 1702 may contains the payment credentials required to perform the contactless payment transaction 908. The payment credentials may include three categories of credentials: common credentials 1706, mag stripe credentials 1708, and chip credentials 1710. The common credentials 1706 may include all data elements common to both mag stripe and chip transactions. The mag stripe credentials 1708 may include data elements specific to mag stripe transactions, and the chip credentials 1710 may include data elements specific to chip transactions. Data elements that may be specific to magnetic stripe transactions or chip transactions will be apparent to persons having skill in the relevant art.

[0235] The PTP 112 may also include the single use key 1704. The single use key 1704 may contain an identifier of the card profile 1702 that it relates to, which may be any type of value suitable for identifying a card profile 1702 stored in the mobile application 1011. In some embodiments, the card profile 1702 and the single use key 1704 may be stored in an encrypted local database 1806 in the data related to the mobile application 1011 in the mobile device 104, such as in data storage 305.

[0236] The single use key 1704 may include an application transaction counter (ATC) and a cryptogram generation key 1712. The cryptogram generation key 1712 may be a single cryptogram generation key to generate one application cryptogram (AC) or one CVC3. The single use key 1704 may be such that it is used to validate a single contactless payment transaction 908, after which time it may be invalid.

[0237] In such an embodiment, dual channel communication may be used to communicate between the cloud (e.g., the payment credentials management system 114) and the mobile application 1011 on the mobile device 104. Dual channel communication may include using both remote notification and a secure socket layer/transport security layer (SSL/TLS) connection.

[0238] FIG. 18 includes an illustration of dual channel communication between the mobile application 1011 and the cloud 114 for use in the present embodiment. The first channel between the mobile application 1011 and the cloud 114 may be a remote notification communication 1802. The remote notification communication 1802 is a communication mechanism that will be apparent to persons having skilled in the relevant art. The remote notification communication 1802 may allow messages or data to be pushed to mobile devices, such as to the mobile device 104 via the mobile application 1011.

[0239] The second channel between the mobile application 1011 and the cloud 114 may be a SSL/TLS communication 1804. The SSL/TLS communication 1804 may allow the establishment of a secure connection between the mobile application 1011 and the cloud 114 used to provide a mutual authentication framework between the mobile application 1011 and the cloud 114. In such a connection, a validation of both the cloud 114 and the mobile application 1011 may be delivered to assure the secured connection.

[0240] In an exemplary embodiment, the dual channel communication may require a shared key and payment credentials (e.g., the PTP 112) to be stored in the mobile application data (e.g., the data storage 305). The mobile application 1011 may generate a local storage encryption key, which may provide the encrypted location database 1806. The local storage encryption key may be based on at least a random value generated by the mobile application 1011, and may be stored in the mobile application data (e.g., the data storage 305).

[0241] FIG. 19 illustrates the present alternative embodiment of the system 100 wherein the mobile device 104 utilizes the card profile 1702 and the single use key 1704 to process mobile payments without using a secure element. The user 113 may register for the service and install the mobile application 1011 on the mobile device 104 using the method as illustrated in FIG. 10 and discussed above. The mobile application 1011 may initialize

during its first execution on the mobile device 104, as illustrated in FIG. 11 and discussed above. In a further embodiment, the user 113 may define a mobile PIN that may be transmitted to the cloud 114 (e.g., in stage [5]) which may be used to access a single use key 1704 for a transaction. In an exemplary embodiment, the mobile PIN may not be stored in the mobile application 1011 or in the mobile device 104.

[0242] The card profile 1702 may be delivered to the mobile device 104 using methods similar to the method illustrated in FIGS. 12A and 12B. In the present embodiment, stage [2] of FIG. 12A may further include the provisioning of the card profile 1702 and any single use keys 1704 stored in the local storage database 1806. Stage [11] may include, in the present embodiment, the building of the card profile 1702. The building of the card profile 1702 may include the retrieval of payment credentials for the user 113, which may include the common credentials 1706 and the mag stripe credentials 1708 and/or the chip credentials 1710, the building of message, generation of an encryption key, and encryption of the message including the card profile 1702. The card profile 1702 may be transmitted to the mobile application 1011 by the cloud 114 in stage [12] as illustrated in FIG. 12A.

[0243] In the present embodiment, once the method illustrated in FIGS. 12A and 12B is completed, then, in stage [22], the mobile application 1011 may not be ready for payment as no single use keys 1704 have been delivered, but may instead be ready to receive single use keys 1704 for the use in future contactless payment transactions 908.

[0244] FIG. 20 illustrates a method for the delivery of a single use key 1704 to the mobile device 104 for the use in a contactless payment transaction 908.

[0245] In stage [1], the cloud 114 may retrieve the encryption key from the user profile used to protect the single use key 1704. The cloud 114 may then build a message including the single use key 1704 using the encryption key and generate a mobile session key used to encrypt the message. In stage [2], the cloud may transmit the encrypted message to the mobile application 1101 using the SSL/TLS connection 1804.

[0246] In stage [3], the mobile application 1101 may generate the same mobile session key, and then, in stage [4], may use the generated key to decrypt the message. The single use key 1704 may then be retrieved by the mobile application 1101 from the message and validated,

and then stored in the local storage database 1806. The mobile application 1101 may then, in stage [5], build a message to notify the cloud of the receipt and validation of the single use key 1704, which may be transmitted by the mobile device 104 in stage[6] using the SSL/TLS connection 1804.

[0247] In stage [7], the cloud 114 may receive the message and decrypt and validate the message, which indicates the receipt and validation of the single use key 1704. In stage [8], the cloud may then activate the single use key 1704 that had been delivered to the mobile application 1101. In some embodiments, the cloud 114 may also inform the issuer 180 of the activated single use key 1704 in stage [9]. The issuer 180 then may, in stage [10], store the received information indicating the active single use key 1704.

[0248] In stage [11], the cloud 114 may notify the mobile device 104 that the single use key 1704 has been activated and instruct the mobile device 104 to wipe the mobile session key. In stage [12], the mobile application 1101 may receive the notification and may update the status of the stored single use key 1704 to active. The mobile application 1101 may then, in stage [13], wipe the mobile session key and mobile storage key, and may, in stage [14], notify the user 113 that the mobile application 1101 is ready to use the single use key 1704 in the contactless payment transaction 908.

[0249] Once the local storage database 1806 includes a single use key 1704, the mobile application 1101 may be used to engage in the contactless payment transaction 908 at the POS terminal 181. The single use key 1704 may generate the AC or the CVC3 key used in the transaction, and the transaction may be processed using the methods and systems as discussed above using the generated AC or CVC3 key.

[0250] A glossary of terms and acronyms described above and depicted in Figures 3-15 and 17-19 is provided in Table 2 below:

Table 2

Symbol/acronym	Description
2FA	Two-factor Authentication
AC	Application Cryptogram
AES	Advanced Encryption Standard

Table 2

Symbol/acronym	Description
AFL	Application File Locator
AID	Application Identifier
AIP	Application Interchange Profile
ATC	Application Transaction Counter
AVN	Application Version Number
BB	BlackBerry
C2DM	Cloud to Device Messaging
CAP	Chip Authentication Program
CP	Card Profile
CBC	Cipher Block Chaining
CNP	Card Not Present
CVC	Card Validation Code
CVC3	Dynamic CVC
DE	Data Element
DES	Data Encryption Standard
DES3	Triple DES
FCI	File Control Information
IMK	Issuer Master Key
IV	Initial Vector
IVS	Initial Vector (Valid for one transaction)
J2ME	Java Platform Micro Edition
JSON	JavaScript Object Notation
KD	Derived Key
KS	Session Key
MSISDN	Mobile Station International Subscriber Directory Number
NATC	Track n Number of ATC Digits
NFC	Near Field Communication
PAN	Primary Account Number
PED	PIN Entry Device
PIN	Personal Identification Number
POS	Point of Sale
PPSE	Proximity Payment System Environment
PSN	PAN Sequence Number
PTP	Payment Token Payload
PUNATC	Track *n* Bitmap for UN and ATC
PCVC3	Track n Bitmap for CVC3
RC	Return Code

Table 2

Symbol/acronym	Description
RFC	Request for Comments
SE	Secure Element
SHA	Secure Hash Algorithm
SIM	Subscriber Identity Module
SMS	Short Message System
SUK	Single Use Key
UDID	Unique Identifier
UDOL	Unpredictable Number Data Object List
UN	Unpredictable Number
⊕	XOR Operator
‖	Concatenation Operator

IV. Exemplary Computer System Implementation

[0251] As would be appreciated by someone skilled in the relevant art(s) and described below with reference to Figure 16, part or all of one or more aspects of the methods and apparatus discussed herein may be distributed as an article of manufacture that itself comprises a computer readable medium having computer readable code means embodied thereon. The computer readable program code means is operable, in conjunction with a computer system, to carry out all or some of the steps to perform the methods or create the apparatuses discussed herein. The computer readable medium may be a recordable medium (e.g., hard drives, compact disks, EEPROMs, or memory cards). Any tangible medium known or developed that can store information suitable for use with a computer system may be used. The computer-readable code means is any mechanism for allowing a computer to read instructions and data, such as magnetic variations on a magnetic media or optical characteristic variations on the surface of a compact disk. The medium can be distributed on multiple physical devices (or over multiple networks). For example, one device could be a physical memory media associated with a terminal and another device could be a physical memory media associated with a processing center.

[0252] The computer systems and servers described herein each contain a memory that will configure associated processors to implement the methods, steps, and functions

disclosed herein. Such methods, steps, and functions can be carried out, e.g., by processing capability on mobile device 104, POS terminal 181, payment processor 103, acquirer 166, issuer 180, or by any combination of the foregoing. The memories could be distributed or local and the processors could be distributed or singular. The memories could be implemented as an electrical, magnetic or optical memory, or any combination of these or other types of storage devices. Moreover, the term "memory" should be construed broadly enough to encompass any information able to be read from or written to an address in the addressable space accessed by an associated processor.

[0253] Aspects of the present disclosure shown in Figures 1-15, or any part(s) or function(s) thereof, may be implemented using hardware, software modules, firmware, tangible computer readable media having instructions stored thereon, or a combination thereof and may be implemented in one or more computer systems or other processing systems.

[0254] Figure 16 illustrates an example computer system 1600 in which embodiments of the present disclosure, or portions thereof, may be implemented as computer-readable code. For example, system 100 of Figures 1, 3-8, 9A, 9B, 12 and 14 and methods and GUI 202 depicted in Figures 2A and 2B can be implemented in computer system 1600 using hardware, software, firmware, non-transitory computer readable media having instructions stored thereon, or a combination thereof and may be implemented in one or more computer systems or other processing systems. Hardware, software, or any combination of such may embody any of the modules and components used to implement the systems and methods described above with reference to Figures 1-20.

[0255] If programmable logic is used, such logic may execute on a commercially available processing platform or a special purpose device. One of ordinary skill in the art may appreciate that embodiments of the disclosed subject matter can be practiced with various computer system configurations, including multi-core multiprocessor systems, minicomputers, mainframe computers, computers linked or clustered with distributed functions, as well as pervasive or miniature computers that may be embedded into virtually any device.

[0256] For instance, at least one processor device and a memory may be used to implement the above described embodiments. A processor device may be a single processor, a plurality of processors, or combinations thereof. Processor devices may have one or more processor "cores."

[0257] Various embodiments of the present disclosure are described in terms of this example computer system 1600. After reading this description, it will become apparent to a person skilled in the relevant art how to implement the present disclosure using other computer systems and/or computer architectures. Although operations may be described as a sequential process, some of the operations may in fact be performed in parallel, concurrently, and/or in a distributed environment, and with program code stored locally or remotely for access by single or multi-processor machines. In addition, in some embodiments the order of operations may be rearranged without departing from the spirit of the disclosed subject matter.

[0258] The processor device 1604 may be a special purpose or a general purpose processor device. As will be appreciated by persons skilled in the relevant art, processor device 1604 may also be a single processor in a multi-core/multiprocessor system, such system operating alone, or in a cluster of computing devices operating in a cluster or server farm. Processor device 1604 is connected to a communication infrastructure 1606, for example, a bus, message queue, network, or multi-core message-passing scheme.

[0259] The computer system 1600 also includes a main memory 1608, for example, random access memory (RAM), and may also include a secondary memory 1610. Secondary memory 1610 may include, for example, a hard disk drive 1612, removable storage drive 1614. Removable storage drive 1614 may comprise a floppy disk drive, a magnetic tape drive, an optical disk drive, a flash memory, or the like.

[0260] The removable storage drive 1614 reads from and/or writes to a removable storage unit 1618 in a well-known manner. The removable storage unit 1618 may comprise a floppy disk, magnetic tape, optical disk, etc. which is read by and written to by removable storage drive 1614. As will be appreciated by persons skilled in the relevant art, the removable

storage unit 1618 includes a non-transitory computer usable storage medium having stored therein computer software and/or data.

[0261] In alternative implementations, the secondary memory 1610 may include other similar means for allowing computer programs or other instructions to be loaded into computer system 1600. Such means may include, for example, a removable storage unit 1622 and an interface 1620. Examples of such means may include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units 1622 and interfaces 1620 which allow software and data to be transferred from the removable storage unit 1622 to computer system 1600.

[0262] The computer system 1600 may also include a communications interface 1624. The communications interface 1624 allows software and data to be transferred between the computer system 1600 and external devices. The communications interface 1624 may include a modem, a network interface (such as an Ethernet card), a communications port, a PCMCIA slot and card, or the like. Software and data transferred via the communications interface 1624 may be in the form of signals, which may be electronic, electromagnetic, optical, or other signals capable of being received by communications interface 1624. These signals may be provided to the communications interface 1624 via a communications path 1626. The communications path 1626 carries signals and may be implemented using wire or cable, fiber optics, a phone line, a cellular/wireless phone link, an RF link or other communications channels.

[0263] In this document, the terms "computer program medium," "non-transitory computer readable medium," and "computer usable medium" are used to generally refer to tangible media such as removable storage unit 1618, removable storage unit 1622, and a hard disk installed in hard disk drive 1612. Signals carried over the communications path 1626 can also embody the logic described herein. The computer program medium and computer usable medium can also refer to memories, such as main memory 1608 and secondary memory 1610, which can be memory semiconductors (e.g. DRAMs, etc.). These computer program products are means for providing software to computer system 1600.

[0264] Computer programs (also called computer control logic and software) are generally stored in a main memory 1608 and/or secondary memory 1610. The computer programs may also be received via a communications interface 1624. Such computer programs, when executed, enable computer system 1600 to become a specific purpose computer able to implement the present disclosure as discussed herein. In particular, the computer programs, when executed, enable the processor device 1604 to implement the processes of the present disclosure, such as the methods illustrated by Figures 2A and 2B, discussed above. Accordingly, such computer programs represent controllers of the computer system 1600. Where the present disclosure is implemented using software, the software may be stored in a computer program product and loaded into the computer system 1600 using the removable storage drive 1614, interface 1620, and hard disk drive 1612, or communications interface 1624.

[0265] Embodiments of the present disclosure also may be directed to computer program products comprising software stored on any computer useable medium. Such software, when executed in one or more data processing device, causes a data processing device(s) to operate as described herein. Embodiments of the present disclosure employ any computer useable or readable medium. Examples of computer useable mediums include, but are not limited to, primary storage devices (e.g., any type of random access memory), secondary storage devices (e.g., hard drives, floppy disks, CD ROMS, ZIP disks, tapes, magnetic storage devices, and optical storage devices, MEMS, nanotechnological storage device, etc.), and communication mediums (e.g., wired and wireless communications networks, local area networks, wide area networks, intranets, etc.).

[0266] Accordingly, it will be appreciated that one or more embodiments of the present invention can include a computer program comprising computer program code means adapted to perform one or all of the steps of any methods or claims set forth herein when such program is run on a computer, and that such program may be embodied on a computer readable medium. Further, one or more embodiments of the present invention can include a computer comprising code adapted to cause the computer to carry out one or more steps of

methods or claims set forth herein, together with one or more apparatus elements or features as depicted and described herein.

V. Conclusion

[0267] While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. It will be apparent to persons skilled in the relevant art that various changes in form and detail can be made therein without departing from the spirit and scope of the invention. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

WHAT IS CLAIMED IS:

1. A method for generating and provisioning payment transaction data to a mobile device having a mobile payment application from a Cloud-based transaction data generation system (Cloud), the method comprising:

provisioning a storage key ($K_{Storage}$), authentication credentials and static payment credentials associated with a payment account to the mobile device, wherein the $K_{Storage}$ key is used to protect static payment credentials stored on the mobile device and the transport of a payload from the Cloud to the mobile payment application;

forwarding the payload comprising at least one of a session key generated using an unpredictable number (KS_{UN}), a dynamic CVC value with Track1/Track2 data ($Cloud\_CVC3_{TRACK1/2}$) and an application transaction counter (ATC) to the mobile payment application, wherein the payload is encrypted prior to the forwarding using the $K_{Storage}$ key;

activating the mobile payment application using a contactless interface in order to enable a contactless payment transaction using the mobile device;

forwarding payment credentials comprising at least one token from the Cloud to the mobile device;

receiving, at the Cloud, a token from a mobile authentication application (MAA) component of the mobile payment application;

validating the token based upon the authentication credentials and at least one additional credential received from the mobile device;

determining, by the Cloud and based on rules, if additional payment credentials need to be provisioned to the mobile device; and

in response to determining that additional payment credentials are needed:

generating the additional payment credentials; and

provisioning the additional payment credentials from the Cloud to the mobile device;

authenticating the payment transaction based on the payment credentials; and

in response to determining that the authenticating was successful, including a genuine CVC3 derived key (KD_{CVC3}) in the payment credentials and returning an encrypted payload to the mobile device including at least one of the KS_{UN}, the $Cloud\_CVC3_{TRACK1/2}$, or the ATC, or in response to determining that the authenticating was unsuccessful including a non-functional KD_{CVC3} key in the payment credentials, returning an encrypted payload to the mobile device without notifying the mobile device of the unsuccessful authentication, and triggering an alarm without notifying the mobile payment application of the unsuccessful authentication,

wherein a secure element in the mobile device is not required.

2. The method of claim 1, wherein the at least one additional credential comprises CAP token, a gesture, a password, passcode, or another suitable Personal Value.

3. The method of claim 1, wherein the validating is performed by an issuer.

4. The method of claim 1, wherein the wherein the validating comprises matching an application transaction counter (ATC) from the Cloud with an ATC received from a merchant or transaction acquirer.

5. The method of claim 1, wherein the transaction data comprises an unpredictable number (UN) used as a seed value as input into a cryptographic process, and wherein the UN is used to compute what an acquirer expects a CVC3 value to be for the transaction.

6. The method of claim 5, wherein the cryptographic process uses the triple Data Encryption Standard (DES) algorithm to generate the CVC3 value and wherein transaction data comprises the CVC3 value.

7. The method of claim 6, wherein a first CVC3 value is a number generated by the Cloud and used by the Mobile Payment Application to generate a CVC3 to perform the payment transaction.

8. The method of claim 7, wherein the payment credentials comprise the dynamic CVC3 cryptogram.

9. The method of claim 1, wherein the token is a Chip Authentication Program (CAP) token indicating one or more controls on purchases.

10. The method of claim 9, wherein the one or more controls limit purchases based upon one or more of:

a day of week;

a time of day;

an expiration date associated with a CAP token;

an expiration date associated with a payment account;

a merchant category corresponding to a point-of-sale (POS) terminal;

a geographic location of a merchant;

a spending limit for a payment account;

a spending limit for a specified merchant category; and

a spending limit for a duration.

11. The method of claim 1, wherein the validating comprises receiving the authentication credentials from the MAA.

12. The method of claim 1, wherein the retrieving comprises synchronizing between the Cloud and an issuer system.

13. The method of claim 1, wherein the Cloud is hosted by an issuer.

14. The method of claim 1, wherein the Cloud is hosted by a third party.

15. The method of claim 14, wherein the third party is a payment processing network.

16. The method of claim 1, further comprising, prior to the processing:

receiving a registration request for a user associated with the mobile device;

processing the registration request;

in response to determining that the registration request has been fulfilled, provisioning the mobile payment application to the mobile device; and

verifying an installation of the mobile payment application on the mobile device.

17. The method of claim 1, wherein the payment account is one or more of:

a credit card;

a debit card;

a pre-paid card;

a hybrid card; or

a payment account with a virtual card number (VCN),

wherein the VCN is a single use VCN, a limited use VCN, or another account number that facilitates a financial transaction using a transaction clearance system.

18. A method for generating and provisioning payment transaction data to a mobile device from a Cloud-based transaction data generation system (Cloud), the method comprising:

authenticating the mobile device to the Cloud based upon authentication credentials previously provisioned to the mobile device;

in response to determining that the authentication is successful, validating, by the Cloud and based on server-side rules, one or more tokens associated with the mobile device;

in response to determining that the mobile device needs new tokens, generating, in the Cloud, one or more new tokens for the mobile device; and

provisioning payment credentials to the mobile device via a mobile payment application previously-provisioned to and installed on the mobile device;

comparing transaction data received from a point-of-sale (POS) with the payment credentials previously-provisioned to the mobile device; and

processing a payment transaction based on the comparing,

wherein a secure element in the mobile device is not required.

19. The method of claim 18, wherein the provisioning of the payment credentials is a pull request from the mobile device.

20. The method of claim 18, wherein the provisioning of the payment credentials is a push from the Cloud to the mobile device.

21. The method of claim 20, wherein the push from the Cloud to the mobile device includes pushing the following parameters/data to a mobile application on the mobile device:

a payment token payload comprising a:

length;

proof information;

payment credentials, wherein the payment credentials are static in the event they are non-sensitive and dynamic in the event they are sensitive; and

an end tag;

an application ID, wherein the application ID is a unique ID used to access a consumer profile for a user associated with the mobile device;

a salt, wherein the salt is a value used in combination with an access code in a the cryptographic process Fn_MA_Key to generate a key used for transport and storage of the payment token payload;

payment parameters comprising:

payment card artwork with a masked PAN value;

a notification URL used to connect to the Cloud to retrieve an encrypted payment token payload; and

a card ID, wherein the card ID is a unique ID used in the generation (Fn_Auth_Code) of the authentication code.

22. The method of claim 18, wherein the authentication is based at least in part on a CAP token, and a received password, passcode, gesture, or Personal Value associated with a user of the mobile device.

23. The method of claim 18, wherein the one or more tokens are based upon one or more merchant categories a payment account associated with the mobile device is authorized for.

24. The method of claim 18, wherein the tokens have time controls on their usage.

25. The method of claim 23, wherein the time controls comprise an expiration date for the one or more new transaction tokens.

26. The method of claim 23, wherein the time controls comprise time of day controls for the tokens.

27. The method of claim 23, wherein the time controls comprise day of week controls for the tokens.

28. The method of claim 3, further comprising verifying, by an issuer, the transaction data.

29. The method of claim 3, wherein the transaction data comprises a dynamic CVC3 cryptogram.

ABSTRACT OF THE DISCLOSURE

A method for providing technical solutions for processing electronic payments initiated from a mobile device without requiring a secure element (SE) in a mobile device, comprising: provisioning authentication credentials and payment credentials associated with a payment account to the mobile device via a pull from the device or a push from a cloud-based transaction data generation system (the Cloud); activating a mobile payment application on the mobile device using a contactless interface to enable a payment transaction using the mobile device; forwarding payment credentials from the Cloud to the mobile device; sending a token from a mobile authentication application (MAA) component of the mobile payment application to the Cloud; validating the token based on upon authentication credentials; determining if additional payment credentials need to be provisioned to the mobile device; and authenticating the contactless payment transaction based on the payment credentials.

MasterCard Worldwide

Remote-SE Mobile *PayPass*
Product Description

Feb 2013 – v3.1.1

© 2013 MasterCard

Proprietary Rights

The information contained in this document is proprietary and confidential to MasterCard International Incorporated, one or more of its affiliated entities (collectively "MasterCard"), or both.

This material may not be duplicated, published, or disclosed, in whole or in part, without the prior written permission of MasterCard.

Trademarks

Trademark notices and symbols used in this manual reflect the registration status of MasterCard trademarks in the United States. Please consult with the Customer Operations Services team or the MasterCard Law Department for the registration status of particular product, program, or service names outside the United States.

All third-party product and service names are trademarks or registered trademarks of their respective owners.

Disclaimer

Access to this Specification is subject to a separate Non-Disclosure Agreement with MasterCard. Implementation of this Specification requires a separate license from MasterCard and may require a license from third party intellectual property owners.

MasterCard makes no representations or warranties of any kind, express or implied, with respect to the contents of this Specification. Without limitation, MasterCard specifically disclaims all representations and warranties with respect to the Specification and any intellectual property rights subsisting therein or any part thereof, including but not limited to any and all implied warranties of title, non-infringement, or suitability for any purpose (whether or not MasterCard has been advised, has reason to know, or is otherwise in fact aware of any information) or achievement of any particular result. Without limitation, MasterCard specifically disclaims all representations and warranties that any practice or implementation of the Specification will not infringe any third party patents, copyrights, trade secrets or other rights.

Table of Contents

Using this Document **1**
- Purpose 1
- Related Information 2
  - *PayPass* Mag Stripe 2
  - *PayPass* M/Chip Requirements 2
  - *Mobile PayPass (MPP)* 2
- Abbreviations 2

Remote-SE Mobile *PayPass* **5**
- Concepts 5
- Limitations 8
- User Experience 9
  - Introduction 9
  - Step 1 10
  - Step 2 10
  - Step 3 11
  - Step 4 11
  - Step 5 12
  - Step 6 13
  - Step 7 13
- Payment Token Payload 15
  - Card Profile (CP) 15
    - Common 15
    - Mag Stripe 16
    - M/Chip 16
  - Single Use Key (SUK) 16
- Dual Channel 17
  - Remote Notification 17
  - Mutual Authentication 18
    - SSL/TLS Communication 18
    - Authentication Code 18
  - Encrypted Local Database in Mobile Payment Application Data 18
- High Level Architecture 19
- Activities 20
  - Overview 20
  - Activity #1 – Registration & Installation 21
    - Overview 21
    - Additional Information 21
  - Activity #2 – Initialization 24
    - Overview 24
    - Additional Information 25
  - Activity #3 – Remote Management 28
    - Activity #3A – Remote Management (Trigger) 29
      - Overview 29
      - Additional Information 29
    - Activity #3B.1 – Remote Management (PTP_CP) 32
      - Overview 32
      - Additional Information 32
    - Activity #3B.2 – Remote Management (PTP_SUK) 34
      - Overview 34
      - Additional Information 34
    - Activity #3B.3 – Remote Management (Mobile_PIN) 36
      - Overview 36
      - Additional Information 36

Activity #3B.4 – Remote Management (Mobile Check) 38
Overview 38
Additional Information 38
Activity #3B.5 – Remote Management (Remote Wipe) 40
Overview 40
Additional Information 40
Activity #4 – Payment (*PayPass* transaction) 42
Overview 42
Additional Information 42

Appendix A – Technical Description 44
Overview 44
Detailed Activity Flows 45
Activity #1 – Registration & Installation 46
Activity #2 – Initialization 47
Activity #3 – Remote Management 48
Activity #3A – Remote Management (Trigger) 49
Activity #3B.1 – Remote Management (PTP_CP) 50
Activity #3B.2 – Remote Management (PTP_SUK) 51
Activity #3B.3 – Remote Management (Mobile_PIN) 52
Activity #3B.4 – Remote Management (Mobile Check) 53
Activity #3B.5 – Remote Management (Remote Wipe) 54
Activity #4 – Payment (*PayPass* transaction) 55
Remote-SE Mobile *PayPass* State Machine (Activities) 56
Key Management 57
Mobile Key 57
Session Key 57
Storage Key 57
Payment Key 58
Encryption Key 58
Functions 59
Fn_Gen_STKey 59
Fn_Gen_MKey 59
Fn_Gen_MSKey 59
Fn_Gen_EK 60
Fn_Gen_RemSEID 60
Fn_Get_RemSEID 60
Fn_Gen_SessID 60
Fn_Gen_RemMgtInfo 60
Fn_Gen_AuthCode 61
Fn_Get_PayCred 61
Fn_Gen_PTPCP 61
Fn_Gen_PTPSUK 61
Fn_Gen_PTP_ActProof 61
Fn_Val_PTP_ActProof 62
Payment Functions 63
Fn_Val_CVC3 63
Fn_Val_AC 63
Fn_Gen_LDA 63
Fn_Gen_SUK 63
Data Elements 64
ST_Key 64
M_Key 64
MS_Key 64
E_Key 64
SUK_Key 64
User_ID 65
RemSE_ID 65
Mobile PIN 65
Mobile_ID 65

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 85

Rnd_Storage 65
Session_ID 65
Activation_Code 65
Authentication_Code 65
PTP_ActProof 65
Initialization_URL 65
Notification_URL 65
RemMgt_Info 65
PTP_CP 65
Common 66
Mag Stripe 66
M/Chip 66
PTPCP_Status 66
PTP_SUK 67
PTPSUK_Status 67
PTPSUK_Counter 67
EPTP_SUK 67

Appendix B – Payment Transaction Flow 68
Overview 68
PayPass Transaction 69
Concepts 69
Transaction Flow 69
Mobile PIN 71
Concepts 71
XOR Method ..... **Error! Bookmark not defined.**
Test Vectors ..... **Error! Bookmark not defined.**
Test Key ..... **Error! Bookmark not defined.**
Mobile PIN (1234) ..... **Error! Bookmark not defined.**
Mobile PIN (98760123) ..... **Error! Bookmark not defined.**
Mobile PIN (0000) ..... **Error! Bookmark not defined.**
Wrong Mobile PIN Entry (54021 i.o.54321) ..... **Error! Bookmark not defined.**
PIN Verification Value 72
Local Data Authentication 72
Need for Mobile CVM 72
LDA Concept 73
Use of Virtual PAN 73

Using this Document

Purpose

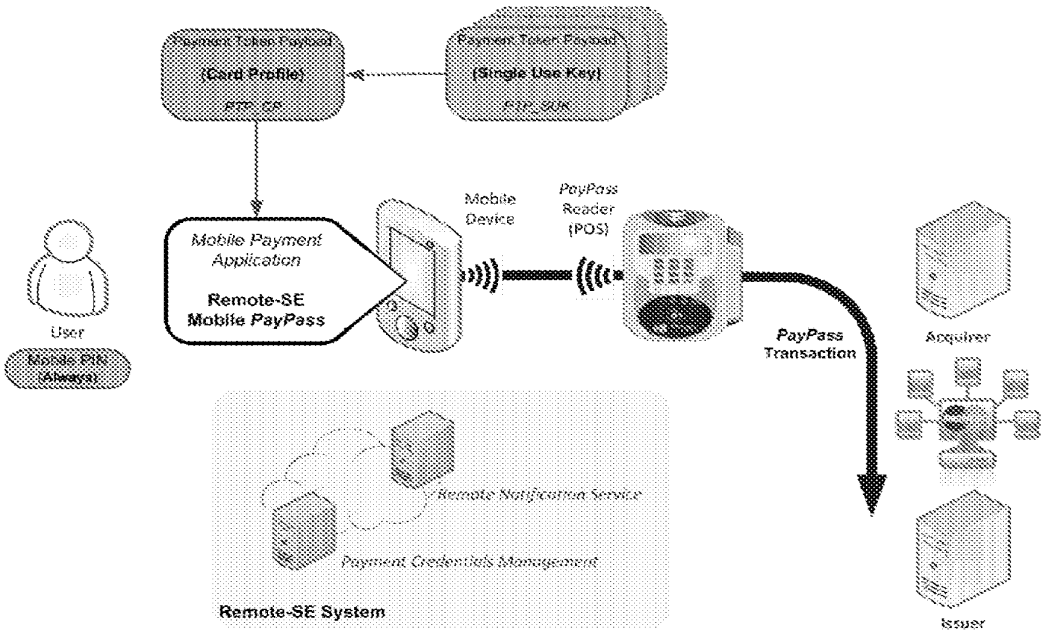
This document describes a solution for *Remote-SE Mobile PayPass*.

It presents the *Concepts* and *Limitations* of the solution.
It shows the *User Experience* when using the Remote-SE Mobile *PayPass* solution to perform *PayPass* transactions (Mag Stripe or M/Chip) with a Mobile Device without using a local Secure Element (SE) personalized with a Payment Application (cardlet).

The solution defines the concepts of *Payment Token Payload* and *Dual Channel*:

- The Payment Token Payload (PTP) is a container used to carry payment credentials from the Remote-SE System to the Mobile Payment Application.
- The Dual Channel combines the use of a remote notification channel with a SSL/TLS connection between the Mobile Payment Application and the Remote-SE System using a mutual authentication (Server authentication provided at communication level [SSL/TLS] and client authentication supported at application level).

The document defines the *High Level Architecture* of the solution.



The document gives a high level definition of the process used to deliver the following *Activities*:

- *Activity #1 – Registration & Installation*
- *Activity #2 – Initialization*
- *Activity #3 – Remote Management*
- *Activity #4 – Payment (PayPass transaction)*

The document contains the following appendices:

- *Appendix A – Technical Description*
  This appendix provides:
  - The detailed description of the activity flows and the state machine operating those activities
  - The list of functions defined to support the various activities
  - The list of data elements used by Remote-SE Mobile *PayPass*
- *Appendix B – Payment Transaction Flow*
  This appendix presents:
  - The Mobile *PayPass* payment transaction flow supported by Remote-SE Mobile *PayPass*
  - The concept of Mobile PIN
  - The concept of Local Data Authentication

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 87

Related Information

***PayPass* Mag Stripe**

[PPMagTS] MasterCard Worldwide, '*PayPass* – Mag Stripe: Technical Specifications', Version 3.3, December 2007

[PPMagIIR] MasterCard Worldwide, '*PayPass* – Mag Stripe: Issuer Implementation Requirements', October 2011

***PayPass* M/Chip Requirements**

[PPReq] MasterCard Worldwide, '*PayPass M/Chip* – Requirements', December 2011

Mobile PayPass (MPP)

[MPP_SU101] MasterCard Worldwide, '*Mobile PayPass: Technical Specifications*', Version 1.0.1, January 2013

Abbreviations

Table 1 contains the abbreviations used in this document.

Table 1—List of Abbreviations

Stage	Description
[M]	Mandatory
[O]	Optional
2FA	Two-factor Authentication
AAC	Application Authentication Cryptogram
AC	Application Cryptogram
ADF	Application Definition File
AES	Advanced Encryption Standard
AFL	Application File Locator
AID	Application Identifier
AIP	Application Interchange Profile
APNS	Apple Push Notification Service
ARQC	Authorization Request Cryptogram
ATC	Application Transaction Counter
AVN	Application Version Number
CAM	Card Authentication Method
CAP	Chip Authentication Program
CBC	Cipher Block Chaining
CDA	Combined DDA/AC Generation
CDOL	Card Risk Management Data Object List
CIAC	Card Issuer Action Code
CID	Cryptogram Information Data
CLD	Card Layout Description
CMK	Card Master Key
CN	Common Name
CNP	Card Not Present
CP	Card Profile
CSK	Common Session Key
CVC	Card Validation Code
CVC3	Dynamic CVC
CVM	Cardholder Verification Method
CVN	Cryptogram Version Number
DDA	Dynamic Data Authentication
DE	Data Element
DES	Data Encryption Standard
DES3	Triple DES
EMV	Europay MasterCard Visa
FCI	File Control Information
GCM	Google Cloud Messaging
HVT	High Value Transaction
IAC	Issuer Action Code
IAD	Issuer Application Data

Stage	Description
ICC	Integrated Circuit Card
IMEI	International Mobile Station Equipment Identity
IMK	Issuer Master Key
IV	Initial Vector
IVCVC3	Initialization Vector for CVC3 generation
J2ME	Java 2 Micro Edition
KD	Derived Key
KS	Session Key
LDA	Local Data Authentication
LVT	Low Value Transaction
MCAL	MasterCard Analysis Laboratory
MPP	Mobile *PayPass*
NATC	Number of ATC Digits
NFC	Near Field Communication
OTP	One Time Password
PAN	Primary Account Number
PCM	Payment Credentials Management
PDOL	Processing Options Data Object List
PED	PIN Entry Device
PKI	Public Key Infrastructure
PIN	Personal Identification Number
PLA	PIN less – Perso less Authentication
PPMS	*PayPass* Mag Stripe
POS	Point of Sale
PPSE	Proximity Payment System Environment
PSN	PAN Sequence Number
PTP	Payment Token Payload
PUNATC	Bitmap for UN and ATC
PVV	PIN Verification Value
RC	Return Code
RFU	Reserved for Future User
SDA	Static Data Authentication
SE	Secure Element
SHA	Secure Hash Algorithm
SMS	Short Message System
SSL	Secure Socket Layer
SU101	Specification Update (v1.0.1)
SUK	Single Use Key
SW	Status Word
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TLV	Tag – Length – Value
TVR	Terminal Verification Results
UDOL	Unpredictable Number Data Object List
UN	Unpredictable Number
URL	Uniform Resource Locator
XOR	Exclusive OR *(Logical Operation)*

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 89

This page is intentionally left blank

IPR2025-01152
Apple EX1008 Page 90

Remote-SE Mobile *PayPass*

Concepts

The core idea of Remote-SE Mobile Payment is to use preloaded payment credentials to perform a standard *PayPass* Mag Stripe or *PayPass* M/Chip transaction between a *PayPass* Reader and a Mobile Payment Application.
The payment credentials are delivered as Payment Token Payloads [Card Profile and Single Use Key] and are stored encrypted (in software) on the Mobile Device. In this representation, a 'Token' does not refer to a physical Token as used in some online (banking) services. It instead refers to a set of data.

- Remote-SE Mobile *PayPass* supports *PayPass* transaction without using a Secure Element (SE) personalized with a Payment Application or a Trusted Execution Environment (TEE) containing a Trusted Application defined to support Mobile Payment.
- Remote-SE Mobile *PayPass* supports Online Only payment transactions (*PayPass* Mag Stripe and *PayPass* M/Chip). The transactions are always authorized online.
- Remote-SE Mobile *PayPass* is a PIN Always solution using a concept of *Mobile PIN*.
  - The Mobile PIN is used to access a payment token (Single Use Key) to support a *PayPass* transaction.
    Both the payment token (Single Use Key) and the related Card Profile have been provisioned to the Mobile Payment Application before the start of the Mobile PayPass transaction.
  - It is not used during the provisioning of payment credentials,
  - It is not verified offline,
  - Never transmitted during a *PayPass* Transaction.
- Remote-SE Mobile *PayPass* does not require any change in the Acceptance Environment.
  - A standard *PayPass* Reader can be used.
  - The solution is compliant with the timing constraints defined for completion of a *PayPass* transaction:
    - *PayPass* Mag Stripe Transaction in less than 250 ms. 170 ms allowed for the Card (in this case a Mobile Payment Application) and 80 ms allowed for the *PayPass* Reader.
    - *PayPass* M/Chip Transaction in less than 500 ms.400 ms allowed for the Card (in this case a Mobile Payment Application) and 100 ms allowed for the *PayPass* Reader.
- v3 Reader supports the concept of Mobile CVM.
- The acceptance of the Remote-SE Mobile *PayPass* is the following for a *PayPass* Reader:

	Hard limit country[1]		**Soft limit** country[2]	
	< v3.0 Reader	v3.x Reader	< v3.0 Reader	v3.x Reader
Transaction Amount **below** CVM limit	*(Technical solution equivalent to Issuer PIN Always)* 1			
Transaction Amount **above** CVM limit	Transaction does not start	Mobile CVM 1	(Mobile CVM) 2 + Online PIN or Signature	Mobile CVM 1

- When using v3 Reader, Mobile CVM prevents "Double PIN entry" (in Mobile and at POS) **[1]**.
  - When using *PayPass* M/Chip, Mobile CVM requires the support of Offline CAM.
  - A concept of *Local Data Authentication* (LDA) is used to deliver Offline CAM support (A CDA like solution) when using Remote-SE Mobile *PayPass*.
  - When using *PayPass* Mag Stripe, Mobile CVM is the standard configuration with Remote-SE Mobile *PayPass*.
- When using < v3.0 Reader, Mobile CVM is not supported by the Reader.
  - For low value transaction, there is one single PIN entry on Mobile **[1]**

[1] [PPReq] (*Hard Limit)* A maximum transaction amount is set and cardholder verification (CVM) is never required below this limit.
[2] [PPReq] (*Soft Limit)* Transactions over a given value require cardholder verification (CVM).

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 91

- ◊ For high value transaction in hard limit countries, the transaction does not start
- ◊ For high value transaction in soft limit countries, the "Double PIN Entry" cannot be avoided.
  There is a PIN entry on Mobile, a Tap of the Mobile Device and then an additional CVM at the POS [illegible].The additional CVM can be Online PIN (PIN Entry on PED) or Signature.

- ♦ Remote-SE Mobile *PayPass* requires limited changes in Issuing/Authorization Environment
  - ◊ When using *PayPass* M/Chip, standard procedures apply using EMV CSK as session key derivation algorithm.
  - ◊ When using *PayPass* Mag Stripe, the Issuer must introduce an additional session key derivation step in the CVC3 validation process (using the same principles as used for *PayPass* M/Chip using EMV CSK as session key derivation algorithm).

- ♦ Remote-SE Mobile *PayPass* uses the concept of *Payment Token Payload*
  - ◊ The Payment Token Payload (PTP) is container used to carry payment credentials from the Remote-SE System to the Mobile Payment Application.
  - ◊ There are two types of Payment Token Payload: *Card Profile (CP)* and *Single Use Key (SUK)*
  - ◊ A Single Use Key is a payment key only valid for one payment transaction (*PayPass* Mag Stripe or *PayPass* M/Chip).
  - ◊ A Card Profile can be linked with a large number of Single Use Keys.
  - ◊ A Single Use Key is always linked to a single Card Profile.

- ♦ Remote-SE Mobile *PayPass* uses the concept of *Dual Channel*:
- ♦ The Dual Channel combines the use of a *Remote Notification* channel with a *SSL/TLS Communication* between the Mobile Payment Application and the Remote-SE System using a *Mutual Authentication* (Server authentication provided at communication level [SSL/TLS] and client authentication supported at application level using an *Authentication Code*).

- ♦ Remote-SE Mobile *PayPass* requires the deployment of a Remote-SE System
- ♦ The Remote-SE System has two main components:
  - ◊ Payment Credentials Management
  - ◊ Remote Notification Service
- ♦ The Remote-SE System has also a connection with the Issuer to synchronize information as part of the monitoring of *PayPass* transactions performed using Remote-SE Mobile *PayPass*.
  (Note: This interface is used to exchange information used by the rules defined to trigger the provisioning of new Single Use Key (SUK))

- ♦ Remote-SE Mobile *PayPass* requires the deployment of a Mobile Payment Application
  - ◊ The Mobile Payment Application is the interface with the User
  - ◊ The Mobile Payment Application manages payment credentials
  - ◊ The Mobile Payment Application supports the *PayPass* Transaction process

- ♦ Remote-SE Mobile *PayPass* defines the concept of *Activities*:
  1. A Registration process allows the User to register to the new service (Remote-SE Mobile *PayPass*),
  2. An Initialization process lets the User perform the activation of the service and initialize a Mobile Payment Application,
  3. A Remote Management process driven by the Remote-SE System is used to manage the Mobile Payment Application,
  4. A Mobile Payment Application provisioned with a Card Profile and at least one Single Use Key can be used to perform a *PayPass* Transaction.

Remote-SE Mobile *PayPass* has also a list of *Limitations* explained in the following section.

Notes

- **Remote-SE Mobile *PayPass* [using a Mobile Payment Application] and Mobile *PayPass* (SE-based) [using a Cardlet] may exist on the same Mobile Device and may be accessed using the same Wallet application:**
  - **The Mobile PIN management will not be same when using Remote-SE Mobile *PayPass* (No offline validation of Mobile PIN) compared to Mobile *PayPass* (Offline validation of the Mobile PIN by the Cardlet)**
  - **This type of integration also assumes that the NFC Chip is able to direct the Card Emulation traffic to either the Secure Element (Mobile *PayPass*) or to the baseband processor (for Remote-SE Mobile *PayPass*) based on one input defined by a Wallet application.**
- **The architecture of Remote-SE Mobile *PayPass* may support multiple instances of Mobile Payment Applications on a single Mobile Device.**

IPR2025-01152
Apple EX1008 Page 93

Limitations

Remote-SE Mobile *PayPass* has the following limitations:

- The use of Remote-SE Mobile *PayPass* with the existing acceptance environment mandates the support of NFC Card Emulation mode to let the Mobile Payment Application access the NFC interface and behave like a contactless card able to communicate with a standard *PayPass* Reader.
- The User must enter a Mobile PIN for every transaction, even for low value transactions (i.e. below the CVM limit).
- Remote-SE Mobile *PayPass* does not support *PayPass* M/Chip Offline Transactions (Note: Offline transactions do not apply to *PayPass* Mag Stripe where transactions are always authorized online).
- The Mobile Device must have connectivity to the Remote-SE System (Using a Wi-Fi or Data connection) to receive remote notification and allow the Mobile Payment Application to connect to the Remote-SE System (SSL/TLS).
- The use of an environment without using a Secure Element (Payment Cardlet) mandates the deployment of several mechanisms to mitigate the risk and provide a solution using payment credentials that can be used one-time in the context of an online authorized transaction.
    - The use of a software-based solution requires added fraud monitoring to detect cloning or relay attacks.
    - The additional fraud detection mechanisms can be supported by both the Issuer and the Remote-SE System.
- The support of EMV (*PayPass* M/Chip) is limited to the generation of an Application Cryptogram (ARQC or AAC) without support of Offline Card Risk Management.
    - The Card Risk Management has to be managed by the Issuer using a control[3] of the provisioning of the payment credentials and the monitoring of Remote-SE Mobile *PayPass* transactions.
    - Any accumulator or counter defined by *PayPass* M/Chip is set to zero when using Remote-SE Mobile *PayPass*.
- Remote-SE Mobile *PayPass* does not support *Torn Transaction* processing, even on v3.x Readers.

[3] Rules are defined by the Issuer to deliver Payment Token Payload (Single Use Keys) using the Remote-SE System.

User Experience

Introduction

The following figure presents the story board describing the user experience of Remote-SE Mobile *PayPass* from the initial registration process to the successful completion of a Mobile *PayPass* transaction.

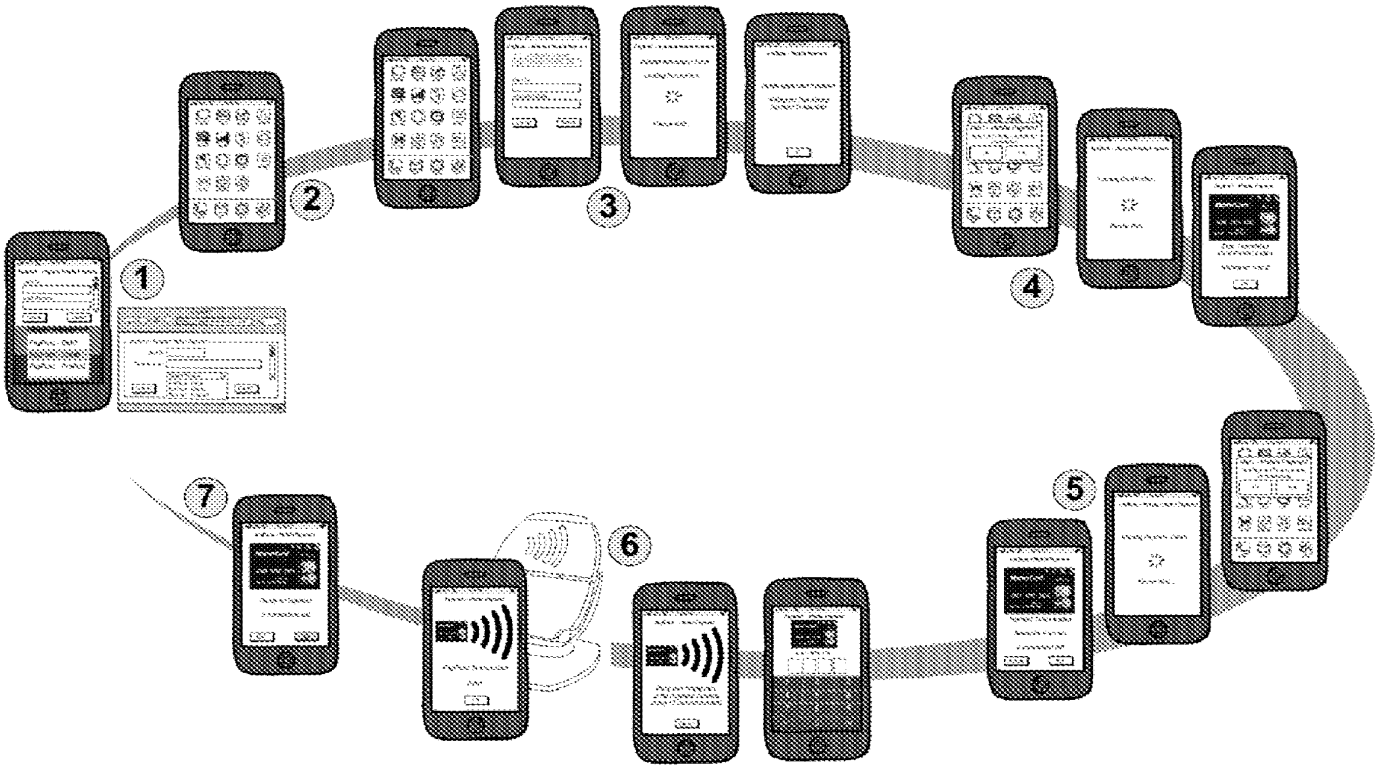


Table 2 describes the steps defined to support the User Experience.

Table 2—User Experience

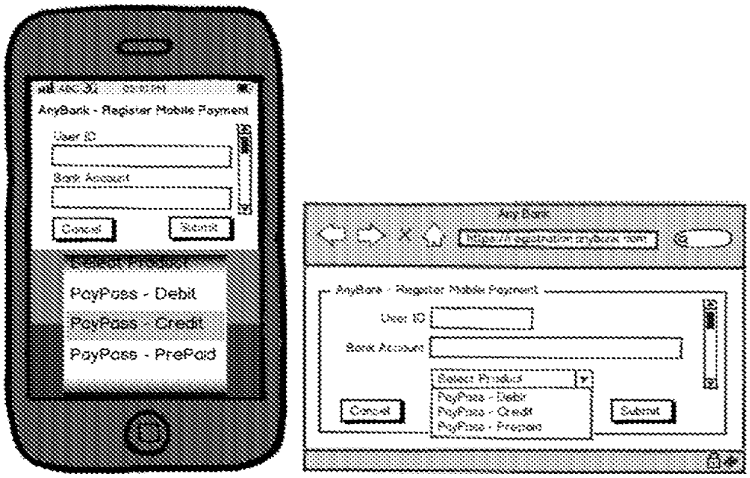
Step	Description
Step 1	User Registration
Step 2	Mobile Payment Application Installation
Step 3	Activation of Mobile Payment Application (= Initialization)
Step 4	Provisioning of Payment Token Payload (Card Profile)
Step 5	Provisioning of Payment Token Payload (Single Use Key)
Step 6	Payment Transaction (*PayPass* Mag Stripe or *PayPass* M/Chip)
Step 7	After a *PayPass* transaction, the User can either perform additional transactions if Single Use Keys (~ Payment Token) are available (go to *Step 6)* or the User needs to have additional Single Use Keys to be pushed and enabled (go to *Step 5*) by the Remote-SE System.

The User Experience described in this section provides a functional description of the processes which will be refined based on usability testing.
Some of the processes (e.g. Payment Token Payload management) may be hidden from the end User.

Step 1

The User registers for the service to receive an *Activation Code*. A User ID known by the User and the Issuer is used to identify the User. It is an Issuer decision to define the value to be used

- The User can use a Mobile-based or a PC-based solution for registration.
- There is no mandate for the registration to use the Mobile Device that will be actually used to install the Remote SE solution (i.e. the Mobile Payment Application).
- The delivery of the Activation Code can be done in one single component delivered using the PC or the Mobile interface.
  It can also be split into several components delivered to the User via several channels (e.g. Face-to-face, using bank statement, by post, using SMS, by email or using any channel usually used between the Issuer and the User).
- The Activation Code is used at time of initialization of the Mobile Payment Application (*Step 3*).



Step 2

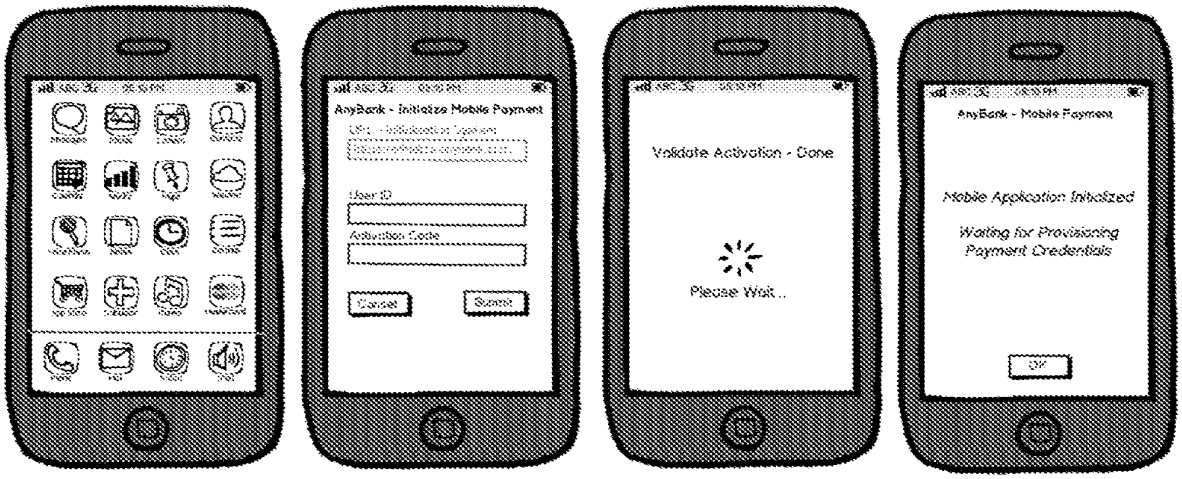
The User has to download the Mobile Payment Application to the Mobile Device that will be used to support Remote SE.
A link to the App Store can be provided to the User as part of the User Registration process described in *Step 1*.
Once the application is installed the User can continue with *Step 3* to initialize the Mobile Payment Application.

Note **The registration (*Step 1*) and the installation (*Step 2*) can be performed in any order. The sequence presented above provides the expected user experience where the User registers first for a service and is guided by the Issuer to download the Mobile Payment Application to be used to support Remote-SE Mobile *PayPass*.**

Step 3

When the User starts the application for the first time, the initialization process must take place.

- The minimum functional requirement is the User to provide a *User ID* (used at time of registration) and the *Activation Code*.
- The initialization process encompasses the creation of a local encrypted database (software) and the registration to remote notification services.
- The local encrypted database uses three factors to build the encryption key:
  - Information retrieved from the Mobile Payment Application
  - Information retrieved from the Mobile Device
  - Random value generated by the Mobile Payment Application
- The Mobile Payment Application will register to the Remote-SE System and a Mobile Key is shared between the Remote-SE System and the Mobile Payment Application.
- Finally the Mobile Payment Application prepares itself for the next steps (*Step 4* – Delivery of Card Profile and *Step 5* – delivery of Payment Token [Single Use Key]).



Step 4

A *PayPass* transaction requires a Card Profile and at least one Payment Token (Single Use Key) to be available.

The Remote-SE System must push a Card Profile to the Mobile Payment Application.

- A remote notification is sent to the Mobile Payment Application. The User is optionally notified that the Mobile Payment Application must connect to the Remote-SE System using a *SSL/TLS Communication*.

IPR2025-01152
Apple EX1008 Page 97

- As part of the use of a *Dual Channel*, a *Mutual Authentication* is performed. When successfully completed, the Card Profile is delivered to the Mobile Payment Application and stored in the local encrypted database.
- There is also an activation process where there is a mechanism for the Mobile Payment Application to inform the Remote-SE System that the Card Profile has been successfully delivered to the Mobile Payment Application.

Once the Card Profile is stored, the Remote-SE System can continue with *Step 5* to push at least one Payment Token (Single Use Key) to the Mobile Payment Application.

Step 5

The delivery of a Payment Token (Single Use Key) to the Mobile Payment Application uses also the *Dual Channel* communication.

- A remote notification is sent to the Mobile Payment Application. The User is optionally notified that the Mobile Payment Application must connect to the Remote-SE System using a *SSL/TLS Communication*.
- The remote notification message contains the Single Use Key encrypted using a random key.
- As part of the use of a *Dual Channel*, a *Mutual Authentication* is performed. When successfully completed, the random key is delivered to the Mobile Payment Application for decryption of the Payment Token (Single Use Key). The Single Use Key is stored in the local encrypted database.
- An activation process is present whereby the Mobile Payment Application informs the Remote-SE System that the Single Use Key has been successfully delivered to the Mobile Payment Application.

Once both a Card Profile and at least one Single Use Key are stored, the Remote-SE System can continue with *Step 6* to proceed with a *PayPass* Transaction.
Additional Payment Tokens may also be pushed to the Mobile Payment Application using the process described in *Step 5*.
The maximum number of Payment Tokens to be delivered will be defined by the Issuer and may also be subject to Issuer defined risk management rules.

The Remote-SE System can also perform various management operations including the replacement of a Card Profile, collect of status information or remote wipe….

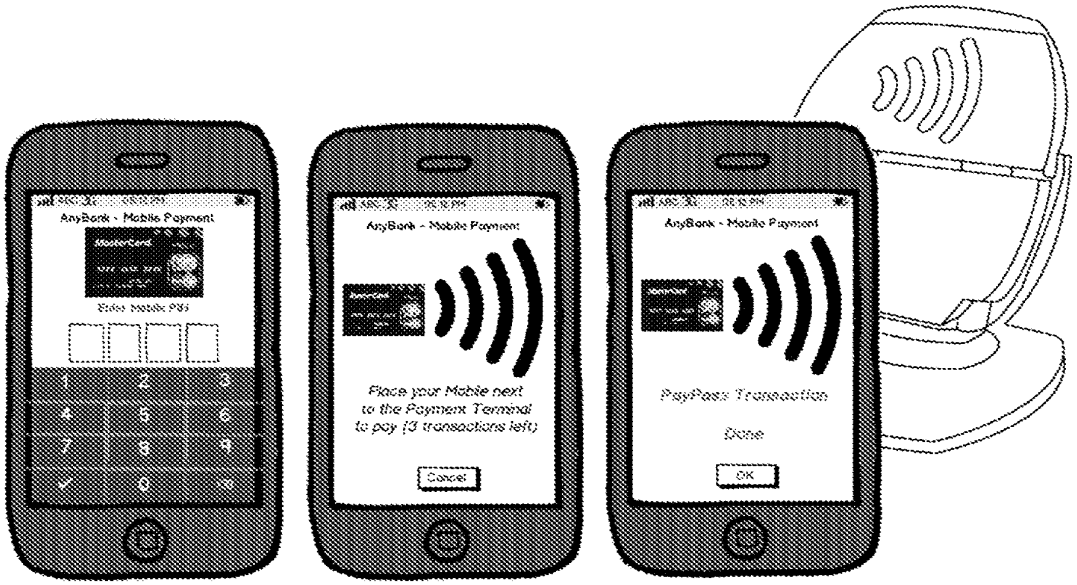
Note **When initializing a Card Profile (and a set of Payment Tokens), the processes described in *Step 4* (PTP_CP) and *Step 5* (PTP_SUK) can be collapsed into one User Experience process (either visible or invisible) even though they are logically separate.**

IPR2025-01152
Apple EX1008 Page 98

Step 6

A *PayPass* transaction requires a Card Profile and at least one Payment Token (Single Use Key) to be available.

- The User must always provide the Mobile PIN for all *PayPass* transactions (Low Value Transaction or High Value Transaction) when using Remote-SE Mobile *PayPass* solution.
- The Single Use Key can only be used once for a single payment transaction.



When the *PayPass* transaction is completed the process described at *Step 7* can be shown to the User.

Step 7

At this stage the Mobile Payment Application is ready for additional *PayPass* transactions.
The Remote-SE System manages remotely the Mobile Payment Application with the following list of features:

- Provisioning of Payment Token (Single Use Key)
- Delivery of a new Card Profile (and removal of any stored Single Use Key(s) associated with the previous Card Profile)
- Management when the Mobile PIN value has been changed (e.g. As a consequence of the modification of the Online PIN value when both values are synchronized)
- Collect of status information (Card Profile and Single Use Key(s)).
- Remote wipe of the content of the Mobile Payment Application.

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 99

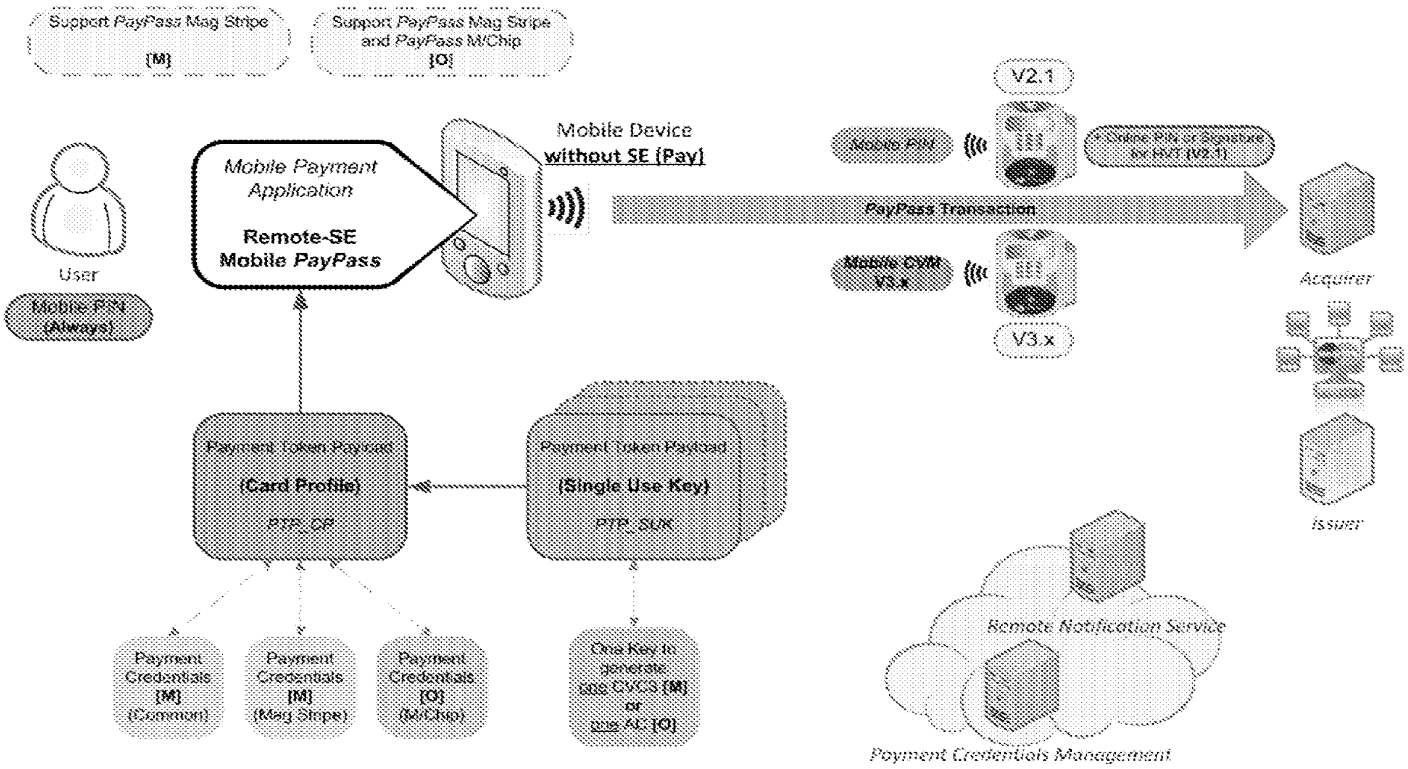
AnyBank - Mobile Payment
MasterCard
XXXX XXXX XXXX
Ready for Payment
2 transactions left
Quit

Payment Token Payload

The Remote-SE Mobile *PayPass* uses the concept of Payment Token Payload (PTP).
A PTP is a container used to carry payment credentials.
There are two types of PTP: *Card Profile (CP)* and *Single Use Key (SUK)*.

The minimum requirement to perform a *PayPass* transaction is to have a Mobile Payment Application initialized and provisioned with a PTP_CP (Card Profile) and at least one PTP_SUK (Payment Token).

- A PTP_CP can be linked with a large number of PTP_SUK.
- A PTP_SUK can only be linked with a single PTP_CP.



The support of *PayPass* Mag Stripe is mandatory **[M]** while the support of *PayPass* M/Chip is optional **[O]**.

Card Profile (CP)

The ***Payment Token Payload – Card Profile (PTP_CP)*** contains the Payment Credentials required to perform a *PayPass* transaction.

The Payment Credentials contained in a Card Profile are split in three categories:

- *Common* – contains all the data elements common to a *PayPass* Mag Stripe and a *PayPass* M/Chip transaction flows.
- *Mag Stripe* – contains all the data elements specific to a *PayPass* Mag Stripe transaction flow
- *M/Chip* – contains all the data elements specific to a *PayPass* M/Chip transaction flow

The PTP_CP does not contain the Application Transaction Counter (ATC) nor the key used to generate cryptogram (CVC3 or AC). That information is contained in a PTP_SUK (*Single Use Key (SUK)*).

Common

The list of *Common* data elements is provided in the *Appendix A – Technical Description* (*PTP_CP – Common*) and includes data elements such as:

- PAN , PSN
- Track Data

- Card Layout Description (CLD) Data

Mag Stripe

The list of *Mag Stripe* data elements is provided in the *Appendix A – Technical Description* (*PTP_CP – Mag Stripe*) and includes data elements such as:

- NATC, PUNATC
- IVCVC3

M/Chip

The list of *M/Chip* data elements is provided in the *Appendix A – Technical Description* (*PTP_CP – M/Chip*) and includes data elements such as:

- CIACs, IACs
- CDOL1, CDOL2
- Application Effective Date, Application Expiration Date
- CVM List
- PDOL
- ICC PKI Information, Issuer PKI Information, Signed Dynamic Application Data

Single Use Key (SUK)

The ***Payment Token Payload – Single Use Key (PTP_SUK)*** is a payment token that can be used one time to generate cryptogram in the context of a *PayPass* transaction.

The *PTP_SUK* contains the Application Transaction Counter (ATC) and the key used to generate cryptogram (CVC3 or AC).

- When using *PayPass* M/Chip, the SUK is the session key (i.e. the session key derivation is already performed by the Remote-SE System before delivery of the key to the Mobile Payment Application).
- When using *PayPass* Mag Stripe, an additional key derivation step is introduced in the CVC3 generation/validation process in order to deliver a solution where the key pushed to the Mobile Payment Application can only be used one time to generate a valid CVC3.
- Remote-SE Mobile *PayPass* uses in both cases the same session key derivation algorithm (EMV CSK session key derivation) where the diversifier is the Application Transaction Counter (ATC).

The PTP_SUK also contains an identifier of the Card Profile (*PTP_CP*) it relates to.

The Remote-SE System uses a mechanism to protect the use of the Single Use Key.
A function (***Error! Reference source not found.***) using the Mobile PIN value is used to protect the access to the correct value of the Single Use Key.

- If a valid Mobile PIN value is provided by the User, the correct value of the key will be used by the Mobile Payment Application.
- If a wrong Mobile PIN value was provided, a key will be used by the Mobile Payment Application but its value is not the expected one.
  Any cryptogram validation performed by the Issuer will fail when authorizing the transaction online.
  - The Mobile Payment Application does not have any support of offline validation of the Mobile PIN value. The mobile application has no other means than doing an online transaction to know whether the Mobile PIN value was correct or not,
  - That way the Issuer controls the Mobile PIN validation process and can apply some fraud detection techniques (using for example a concept of Mobile PIN Try Counter managed at Issuer level).

Detailed information about *Mobile PIN* is provided in *Appendix B – Payment Transaction Flow.*

Dual Channel

The Remote-SE Mobile *PayPass* solution uses a Dual Channel for communication between the Remote-SE System and the Mobile Payment Application:

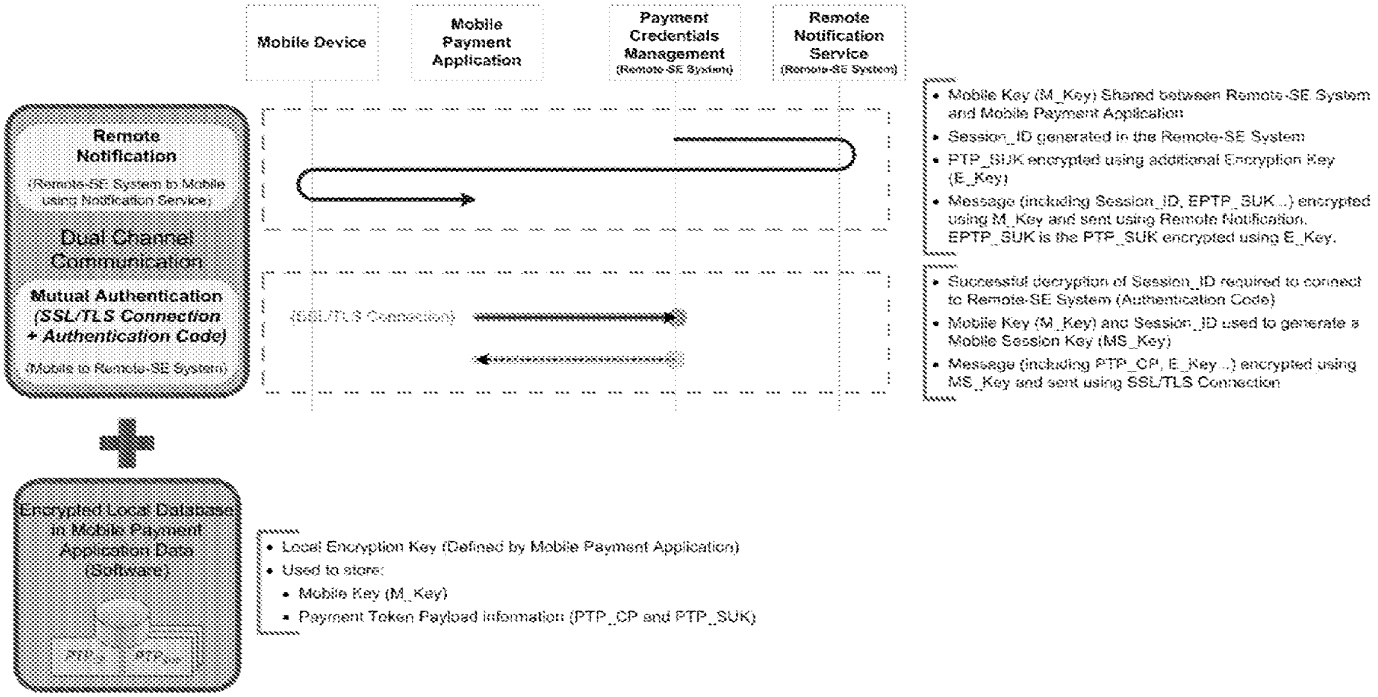
- *Remote Notification*
- *Mutual Authentication*
  - *SSL/TLS Communication*
  - *Authentication Code*

The solution also uses an *Encrypted Local Database in Mobile Payment Application* Data.

The Remote-SE System encompasses:

- Payment Credentials Management
- Remote Notification Service

The figure below gives an overview of the architecture and lists the services provided by the components involved in the Dual Channel communication framework.



Remote Notification

The Mobile Payment Application must register to a Remote Notification service during the initialization of the application on the Mobile Device (Refer to *Activity #2 – Initialization*).

The Remote Notification mechanism is a standard process available for the following development platforms:

- Android (Google Cloud Messaging)
- Apple (iOS)
- BlackBerry
- J2ME
- Windows Phone

The mechanism allows messages to be pushed to Mobile Devices (previously registered) deployed in the field. The Mobile Payment Application does not need to be running to receive messages.

Remote-SE Mobile *PayPass* solution uses its own mechanisms (e.g. using encryption with a Mobile Key shared between the Mobile Payment Application and the Remote-SE System) to protect the confidentiality of data exchanged using the Remote Notification Service.

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 103

Mutual Authentication

Remote-SE Mobile *PayPass* provides mutual authentication between the Mobile Payment Application and the Remote-SE System using:

- Server authentication – *SSL/TLS Communication*
- Client authentication – *Authentication Code*

SSL/TLS Communication

The SSL/TLS is standard mechanism to allow the establishment of a secure connection between the Mobile Payment Application and the Services hosted in the Remote-SE System.
The Mobile Payment Application validates the Server Certificate checking the Common Name (CN) against a predefined value using information from the Mobile Payment Application (and from the Mobile Device using a list of trusted certificates).

Authentication Code

The Authentication code is a hash computed over a set of data known by the Mobile Payment Application and the Remote-SE System:

- A unique identifier (*RemSE_ID*) defined by the Remote-SE System to uniquely identify a Remote-SE User Profile. The value is provided to the Mobile Payment Application at time of initialization of the application (*Activity #2 – Initialization*).
- A session identifier (*Session_ID*) delivered by the Remote-SE System as part of the remote notification message (*Activity #3A – Remote Management (Trigger)*).
  The value is sent encrypted and the Mobile Payment Application must have access to the encrypted local database to retrieve the Mobile Key (*M_Key*) used by the Remote-SE System to encrypt the data.

The Authentication Code provides the assurance that an identified (using *RemSE_ID*) Mobile Payment Application has received a remote notification message and was able to decrypt its content to retrieve a session identifier (*Session_ID*).

Encrypted Local Database in Mobile Payment Application Data

The Dual Channel mechanism requires a shared key (the Mobile Key), Payment Credentials and parameters to be stored in the Mobile Payment Application data.
A Local Storage Encryption key (*ST_Key*) is generated by the Mobile Payment Application using three factors:

- Information from the Mobile Device.
  The Mobile Payment Application can use the IMEI value (when using a Mobile Device) or a Serial Number (when using a Tablet without telephony support).
- Information from the Mobile Payment Application Code.
  The value is defined by the owner of the Mobile Payment Application and is usually obfuscated in the source code of the application.
- A random value (*Rnd_Storage*) generated by the Mobile Payment Application.
  The value is generated during the initialization of the application (Activity #2 – Initialization) and must be stored in the Mobile Payment Application data.

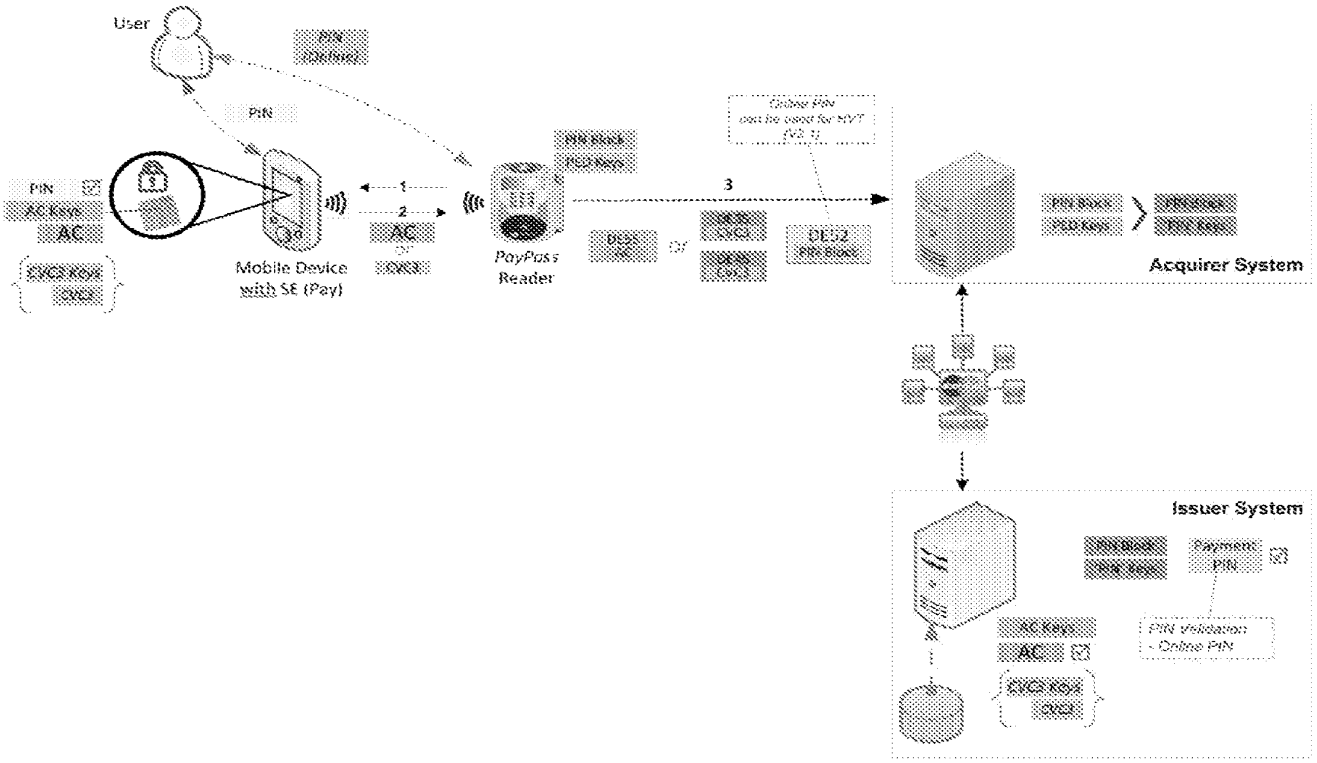
The Mobile Payment Application uses this key to provide an encrypted local database[4] used to store the information provided by the Remote-SE System:

- Mobile Key (*M_Key*)
- Payment credentials (*PTP_CP* and list of *PTP_SUK*)
- Status and counter information (*PTPCP_Status*, *PTPSUK_Status* and *PTPSUK_Counter*)
- Parameters (*RemSE_ID* and *Notification_URL*).

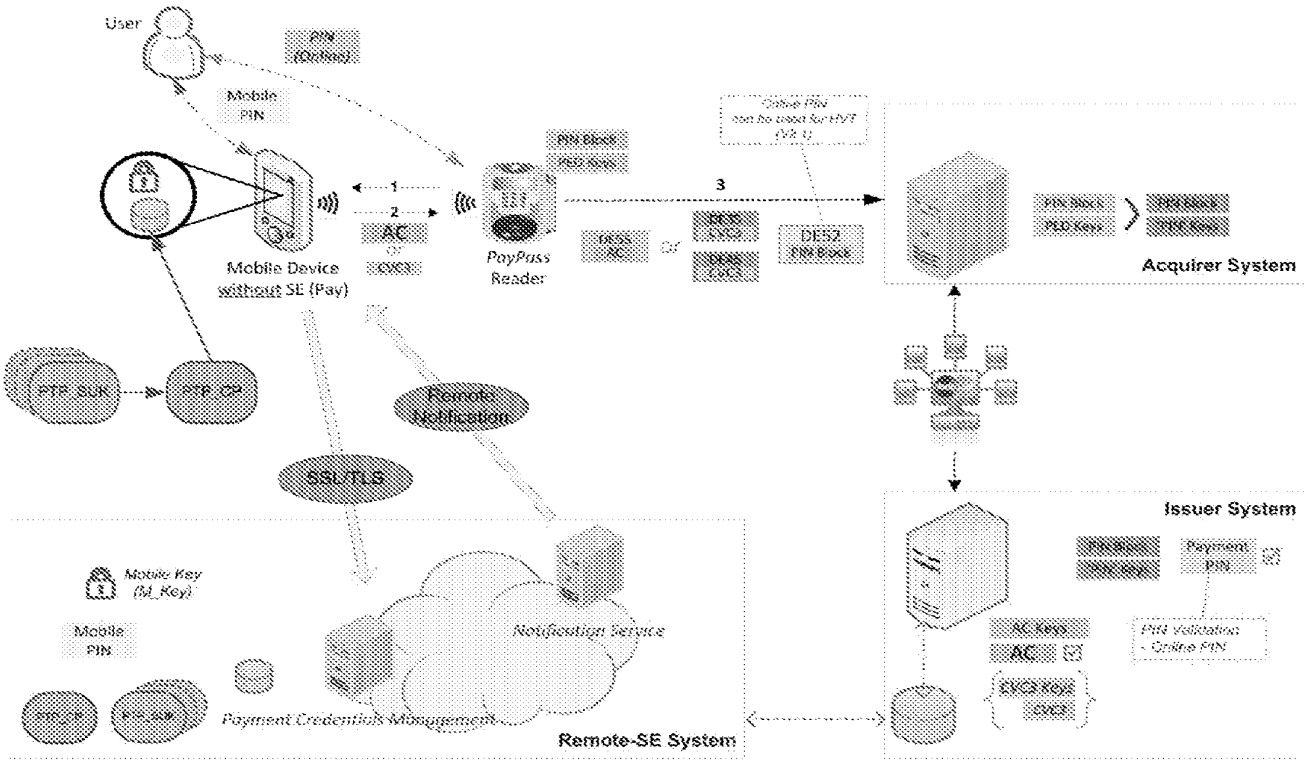
[4] The concept of "database" means a mechanism allowing data to be managed (Add, Delete, Update and Search) by the Mobile Payment Application. The confidentiality of the data is provided using an encryption mechanism using a storage key (*ST_Key*).

High Level Architecture

The following figure describes a standard SE-based Mobile *PayPass* payment transaction architecture:



The Remote-SE Mobile *PayPass* solution supports the same type of Mobile *PayPass* transaction but using a Mobile Device without a SE personalized for Payment.



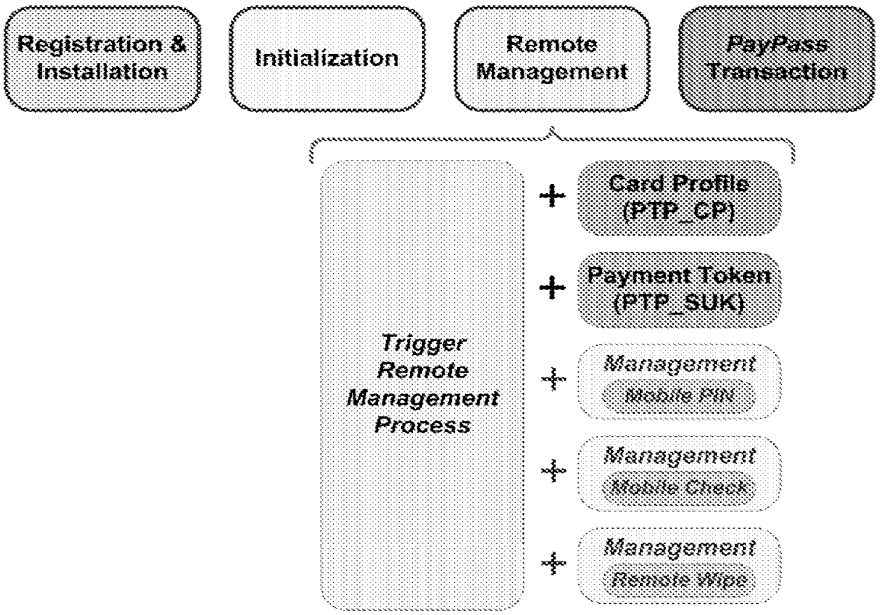
Activities

This section presents the activities defined to support the various processes of Remote-SE Mobile *PayPass* solution.

Overview

There are four main activities defined for Remote-SE Mobile *PayPass*:

- *Activity #1 – Registration & Installation*
- *Activity #2 – Initialization*
- *Activity #3 – Remote Management*
- *Activity #4 – Payment (PayPass transaction)*



The management of activities for Remote-SE Mobile *PayPass* is the following:

- *Activity #1 – Registration & Installation* is a pre-requisite to start *Activity #2 – Initialization*.
- Upon successful completion of *Activity #2 – Initialization* the Remote-SE System can proceed with remote management of the Mobile Payment Application.
- The Mobile Payment Application requires at least a Card Profile (PTP_CP) and a Payment Token (PTP_SUK) to allow the User to perform a Mobile *PayPass* Transaction.
  - The provisioning of Payment Credentials must always start with the delivery of Payment Token Payload (Card Profile) using the *Activity #3B.1 – Remote Management (PTP_CP)*.
  - When a Card profile is available, the provisioning of Payment Token Payload (Single Use Key) using the *Activity #3B.2 – Remote Management (PTP_SUK)* can be performed.
  - Both PTP_CP and PTP_SUK activities share a common process to trigger the Remote Management as defined in *Activity #3A – Remote Management (Trigger)*.
- Additional activities are also defined for remote management of the Mobile Payment Application (*Activity #3 – Remote Management*).
  Those activities also share the common process to trigger the Remote Management.
- Once the provisioning of Payment Credentials is successfully completed, the User can execute the *Activity #4 – Payment (PayPass transaction)* to perform a Mobile *PayPass* transaction using the Mobile Payment Application.

Activity #1 – Registration & Installation

This section provides first an *Overview* of the activity and then provides *Additional Information*.

The detailed description of the flow used to support *Activity #1 – Registration & Installation* is provided in the *Appendix A – Technical Description*.

Overview

The ***Registration and Installation*** activity uses the following components:

- App Store
- Browser (PC or Mobile) (or any Issuer application used to support the registration process)
- User
- Mobile Device
- Mobile Payment Application
- Remote-SE System (Payment Credentials Management)
- Issuer[5]

The process supported by this activity aims to:

A. Let the User access a registration system in the Cloud (directly connecting to the Remote-SE System or using the Issuer online banking system to access that service),
B. Use some Issuer defined mechanisms to identify and authenticate the User,
C. Provide guidance to the User to download and install a Mobile Payment Application from an App Store,
D. Validate a request from the User to enable Remote-SE Mobile *PayPass* for a given bank account,
E. Create a Remote-SE User Profile which can be uniquely identified,
F. Deliver an Activation Code (*Activation_Code*) required to initialize the Mobile Payment Application (*Activity #2 – Initialization*),
G. Optionally ask the User to define the value of the Mobile PIN.
H. Support Fraud Management

At the end of the process, the status is the following:

- The Mobile Payment Application is installed on the Mobile Device that will be used to support Mobile *PayPass* transactions using Remote-SE Mobile *PayPass* solution.
- A Remote-SE User Profile has been created in the Remote-SE System.
  The profile can be identified using a unique identifier *RemSE_ID* (linked with the User profile identified using *User_ID*) defined by the Remote-SE System.
- An Activation Code is stored in that profile and will be used to grant access to the initialization of the Mobile Payment Application (*Activity #2 – Initialization*).

Additional Information

A. Registration system
  - The registration system can leverage from the existing interfaces between the Issuer and the User (e.g. online banking system, Issuer wallet application…).
  - There is no technical requirement to perform the registration process using the Mobile Device that will be used to install the Mobile Payment Application.
  - The access to the registration system is done using a SSL/TLS connection with server authentication using a validation of the Common Name by the User (using some standard guidance from the Issuer).
  - The registration process is commonly initiated using a browser (PC, Mobile, Tablet or TV) but it may also use an Issuer application.

[5] Including a communication channel between the Remote-SE System and the Issuer to synchronize and share information.

IPR2025-01152
Apple EX1008 Page 107

B. Identification and authentication

- The solution assumes that the User and the Issuer (Remote-SE System) share one value (*User_ID*) which can be used to uniquely identify the User. This identifier may optionally identify a payment product used by the User (e.g. when using a PAN as an identifier).
- The authentication process may leverage from the strong authentication mechanisms used by the Issuer for online banking or any access to Cloud-based services.
- The strong authentication can use some Two-Factor Authentication (2FA) solutions such as Chip & PIN (using a chip card and a personal card reader), Display Card, SMS, OTP Generator or any other Issuer defined mechanism.

C. App Store

- The App Store is a repository of mobile applications which can be accessed using the Mobile Device.
- The Remote-SE System can provide a link to the App Store to ease the download of the Mobile Payment Application to the Mobile Device (or for PC-based registration providing a QR Code containing a URL to directly download the mobile application from the App Store).
- The Mobile Device environment can provide standard mechanisms to guarantee the integrity of the Mobile Payment Application and provide a means to identify (and authenticate) the owner (or author) of the mobile application.
- The security level of the installation and validation process of Mobile Payment Application relies on the standard solutions provided by the Mobile Vendors (e.g. Apple Store, Google Play…).

D. User Request

- The authentication framework used to control the access to the registration system (Login) can also be used to provide some mechanism to authenticate the User request to enable the Remote-SE Mobile *PayPass* for a selected bank account.
- A "signature" process can be used for that purpose (refer to the concept of authentication mode as defined in *MasterCard Authentication Solutions* using CAP or PLA technologies).

E. Remote-SE User Profile

- A Remote-SE User Profile is created in the Remote-SE System at time of the registration of the User.
- The profile is used to store all the Remote-SE related information for that User and the Remote-SE Mobile *PayPass* instance used by that User.
- This profile is uniquely identified using *RemSE_ID*. This identifier can be also retrieved by the Remote-SE System using the User identifier (*User_ID*).

F. Activation Code

- The Activation Code (*Activation_Code*) is a link between the activities *Activity #1 – Registration & Installation* and *Activity #2 – Initialization.*
- The value of the Activation Code is generated by the Remote-SE System.
- The mechanisms used to generate and validate that value are out of scope of this product description but they must guarantee that the value can only be used once (= concept of One-Time Password).
- The Activation Code can be delivered to the User using the channel used to register to the service or it can be done using another channel (e.g. Bank Statement, SMS, Email, Face-to-face delivery or any other communication channel between the Issuer and the User).
- It is an Issuer (Remote-SE System) decision to split the Activation Code into several components (to be delivered using several channels).
  When this optional feature is used, the Mobile Payment Application must provide a means for the User to enter all the components at time of the initialization of the application (*Activity #2 – Initialization*)

G. Mobile PIN

- ◊ The default value for the *Mobile PIN* is the Online PIN.
- ◊ The Mobile PIN is never stored in the Mobile Payment Application data.
- ◊ The Mobile PIN is never validated by the Mobile Payment Application.
- ◊ The Mobile PIN is never transmitted during a *PayPass* Transaction
- ◊ It is an Issuer decision to provide the option for the User to define the Mobile PIN during the registration process (or in the context of the *Activity #2 – Initialization*)
- ◊ The User defined Mobile PIN value may also be part of the "signature" process that can be used by the Remote-SE System (Issuer) to validate the User Request to register to Remote-SE Mobile *PayPass*.

H. Fraud Management

- ◊ The User is identified and registered to support Remote-SE Mobile *PayPass.*
- ◊ The User has received an Activation Code.
- ◊ No *PayPass* transaction is allowed at this stage for the selected product.

Activity #2 – Initialization

This section provides first an *Overview* of the activity and then provides *Additional Information*.

The detailed description of the flow used to support *Activity #2 – Initialization* is provided in the *Appendix A – Technical Description*.

Overview

The ***Initialization*** activity uses the following components:

- User
- Mobile Device
- Mobile Payment Application
- Remote-SE System (Payment Credentials Management)
- Remote-SE System (Remote Notification Service)
- Issuer

The process supported by this activity aims to:

A. Let the User start the Mobile Payment Application for the first time,

B. Let the Mobile Payment Application ask the User to provide an identifier (*User_ID*) and the Activation Code (*Activation_Code*) required for initialization of the Mobile Payment Application,

C. Initialize the Mobile Payment Application
  - Generation of a Mobile Identifier (*Mobile_ID*) during registration to Remote Notification Services
  - Generation of a storage key (*ST_Key*) used to encrypt a local database

D. Connect the Mobile Payment Application to the Remote-SE System (using *Initialization_URL*) and deliver the initialization credentials:
  - User Identifier (*User_ID*)
  - Activation Code (*Activation_Code*)
  - Mobile Identifier (*Mobile_ID*)

E. Validate the Activation Code and complete registration of the Mobile Payment Application (and the Mobile Device) to the Remote Notification Service,

F. Generate a Mobile Key (*M_Key*) that will be shared between the Remote-SE System and the Mobile Payment Application,

G. Populate the Remote-SE User Profile with information,

H. Deliver information back to the Mobile Payment Application and store them in the local encrypted database:
  - Mobile Key (*M_Key*)
  - Remote-SE Identifier (*RemSE_ID*)
  - URL (*Notification_URL*) to be used to access the Remote-SE System upon reception of a remote notification message

I. Optionally ask the User to define the value of the Mobile PIN.

J. Support Fraud Management.

At the end of the process, the status is the following:

- The Mobile Payment Application (on the Mobile Device) is initialized and ready to support Remote Management (*Activity #3 – Remote Management*).
- A Mobile Key (*M_Key*) is shared between the Mobile Payment Application and the Remote-SE System.
- A local encrypted database is available on the Mobile Device and the Mobile Payment Application uses it to store parameters (and payment credentials to be provisioned using the remote management activities).

Additional Information

A. Application start

- The initialization process must be initiated using the Mobile Device that will be used to support the Remote-SE Mobile *PayPass* "card".
  The current solution assumes that a given Remote-SE Mobile *PayPass "card"* cannot be loaded on several Mobile Devices.
- The Mobile Payment Application performs some integrity checks at startup and detection for rooted device.
- *The security model assumes that there is no 100% guarantee that the Mobile Payment Application is able to detect a rooted device or detect that the device had been previously rooted and was "unrooted" afterwards.*

B. Identification and authentication

- As stated in the *Activity #1 – Registration & Installation*, the solution assumes that the User and the Issuer (Remote-SE System) share one value (*User_ID*) which can be used to uniquely identify the User. That value was used at time of registration of the User to the Remote-SE System.
- The authentication process may leverage from the strong authentication mechanisms used by the Issuer for online banking or any access to Cloud-based services.
- The strong authentication can use some Two-Factor Authentication (2FA) solutions such as Chip & PIN (using a chip card and a personal card reader), Display Card, SMS, OTP Generator or any other Issuer defined mechanism.
- The same way, the authentication framework used to control the access to the registration system can also be used to provide some mechanism to authenticate the User request to initialize the Mobile Payment Application.
  A "signature" process can be used for that purpose (refer to the concept of authentication mode as defined in *MasterCard Authentication Solutions* using CAP or PLA technologies).

C. Initialize Mobile Payment Application

- The Mobile Payment Application must register to the Remote Notification Service.
  A unique identifier (*Mobile_ID*) is generated as part of that process <u>using standard mechanisms (APIs)</u> delivered by Mobile Vendors (e.g. Google Cloud Messaging for Android (GCM), Apple Push Notification Service (APNS)…).
  This type of service guarantees that a Remote Notification message can be sent to a given Mobile Payment Application installed on a given Mobile Device.
- The Mobile Payment Application must generate a random value (*Rnd_Storage*) used to build a storage key (*ST_Key*) used to encrypt a local database as defined in *Encrypted Local Database in Mobile Payment Application Data* section.

D. Connection to Remote-SE System

- The Mobile Payment Application uses a default URL[6] to connect to the Remote-SE System.
- The mobile application validates the SSL/TLS connection (Certificates) and checks the Common Name (CN) against a stored value[7].
- Another URL (and reference Common Name) is delivered to the Mobile Payment Application as part of the initialization process.
  The URL (*Notification_URL*) is used to access the Remote-SE System upon reception of a remote notification message.
- The URL is stored by the Mobile Payment Application in the local encrypted database.

E. Activation Code

- The Activation Code (*Activation_Code*) is a link between the activities *Activity #1 – Registration & Installation* and *Activity #2 – Initialization.*
- The value of the Activation Code was generated by the Remote-SE System and delivered to the User during the registration process.
- The Remote-SE System identifies the User (*User_ID*) and validates the Activation Code against a value stored in the Remote-SE User Profile.
- The Activation Code can only be used one-time.
  The Remote-SE System must update the Remote-SE User Profile accordingly.

---

[6] The URL (*Initialization_URL*) is stored in Mobile Payment Application source code

[7] That value is also part of the object (*Initialization_URL*) stored in the Mobile Payment Application source code.

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 111

F. Mobile Key

- The Mobile Key (*M_Key*) is generated by the Remote-SE System.
- The value is sent to the Mobile Payment Application using the SSL/TLS connection used during the initialization activity.
- The key is stored by the Mobile Payment Application in the local encrypted database.

G. Remote-SE User Profile

- The Remote-SE User Profile was created in the Remote-SE System at time of the registration of the User.
- The profile contains all the Remote-SE related information for that User and the Remote-SE Mobile *PayPass* instance used by that User.
- The profile is uniquely identified using *RemSE_ID*. This identifier can be also retrieved by the Remote-SE System using the User identifier (*User_ID*).
- The Remote-SE User ID is part of the information sent back to the Mobile Payment Application. The value is stored by the Mobile Payment Application in the local encrypted database.

H. Local Encrypted Database

- The local encrypted database is used by the Mobile Payment Application to stored several parameters (*M_Key*, *RemSE_ID* and *Notification_URL*).
- The random value (*Rnd_Storage*) cannot be stored in the local encrypted database. It must be stored in the Mobile Payment Application data to allow the Mobile Payment Application to regenerate the storage key (*ST_Key*) when access is required to the local encrypted database.
- It should be noted that the storage key (*ST_Key*) could be regenerated from a memory dump or eavesdropped, hence the Mobile key (*M_Key*) could be determined and the system compromised for that instance of Remote-SE Mobile *PayPass*.

I. Mobile PIN

- The default value for the *Mobile PIN* is the Online PIN.
- It is an Issuer decision to provide the option for the User to define the Mobile PIN during the registration process (*Activity #1 – Registration & Installation*) or in the context of this activity.
- The User defined Mobile PIN value may also be part of the "signature" process that can be used by the Remote-SE System (Issuer) to validate the User Request to initialize to Remote-SE Mobile *PayPass*.
- As stated in *Activity #1 – Registration & Installation*, the Mobile PIN is never stored in the Mobile Payment Application data and the Mobile PIN is never validated by the Mobile Payment Application.

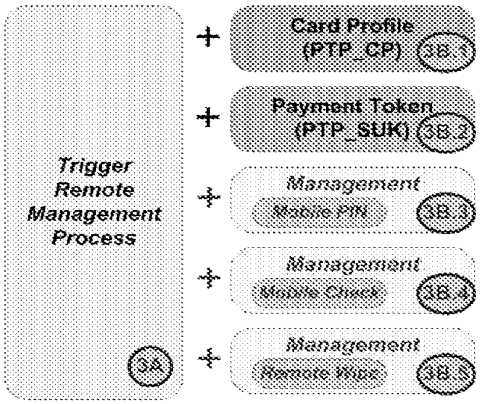
J. Fraud Management

- The Mobile Payment Application is ready to be remotely managed.
- Rules are defined to support the remote management of Payment Credentials (Card Profile and Single Use Keys)
- No *PayPass* transaction is allowed at this stage for the selected product.

Activity #3 – Remote Management

When using Remote-SE Mobile *PayPass*, the Remote-SE System (Payment Credentials Management) is able to perform remote management of the Mobile Payment Application (running on the Mobile Device) using a Remote Notification mechanism.
The *Remote Notification* is a standard component of Mobile OS which allows a Server to **Push** messages to a Mobile Payment Application previously registered.

- The Remote Notification triggers a process at level of the Mobile Device.
  - *Activity #3A – Remote Management (Trigger)*
    The Remote Notification process is common to all the Remote Management activities.
- From that point, the Mobile Payment Application is started and connects to the Remote-SE System (Payment Credentials Management) using a *SSL/TLS Communication* layer with *Notification_URL*.



- Once successfully authenticated (*Mutual Authentication*) as part of the remote management trigger process (*Activity #3A – Remote Management (Trigger)*) , the management of the Mobile Payment Application can be performed using one of the following activities:
  - *Activity #3B.1 – Remote Management (PTP_CP)*
    Management of Payment Token Payload (Card Profile)
  - *Activity #3B.2 – Remote Management (PTP_SUK)*
    Management of Payment Token Payload (Single Use key)
  - *Activity #3B.3 – Remote Management (Mobile_PIN)*
    Management of Mobile Payment Application when Mobile PIN has been changed (e.g. following the change of Online PIN if this value is used as the value of Mobile PIN). In this case, all the preloaded Payment Token Payload (Single Use Key) must be removed from the Mobile Payment Application.
  - *Activity #3B.4 – Remote Management (Mobile Check)*
    Function to retrieve the current status of the Payment Token Payload (PTP_CP and PTP_SUK) available in the Mobile Payment Application.
  - *Activity #3B.5 – Remote Management (Remote Wipe)*
    This function is designed to address fraud issues or any termination of service.
    This function wipes all the information stored in the Mobile Payment Application (i.e. Payment Credentials, Mobile Key and any other stored values) while providing the status of the Mobile Payment Application before the removal of credentials is executed by the Mobile Payment Application.

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 114

Activity #3A – Remote Management (Trigger)

This section provides first an *Overview* of the activity and then provides *Additional Information*.

The detailed description of the flow used to support *Activity #3A – Remote Management (Trigger)* is provided in the *Appendix A – Technical Description*.

Overview

The ***Remote Management (Trigger)*** activity uses the following components:

- User
- Mobile Device
- Mobile Payment Application
- Remote-SE System (Payment Credentials Management)
- Remote-SE System (Remote Notification Service)
- Issuer

The process supported by this activity aims to:

A. Trigger the remote management of a Mobile Payment Application using either a rule defined in the Payment Credentials Management (Remote-SE System) or using a trigger process performed by the Issuer,

B. Generate a session identifier (*Session_ID*) that will be used to support the remote management operation,

C. Prepare data for Remote Notification message,

D. Deliver the notification message to the Mobile Payment Application using standard Remote Notification Services,

E. Activate the Mobile Payment Application and validate its integrity,

F. Get access to the parameters required to decrypt and validate the content of the remote notification message,

G. Connect the Mobile Payment Application to the Remote-SE System (using *Notification_URL*) and deliver the authentication credentials:
  - Identifier (*RemSE_ID*)
  - Authentication Code (*Authentication_Code*)

H. Validate the Authentication Code and continue the process defined for the selected remote management operation.

When the process is completed, the Mobile Payment Application is ready to process the second part of the Remote Management defined by the function to be used:

- *Activity #3B.1 – Remote Management (PTP_CP)*
- *Activity #3B.2 – Remote Management (PTP_SUK)*
- *Activity #3B.3 – Remote Management (Mobile_PIN)*
- *Activity #3B.4 – Remote Management (Mobile Check)*
- *Activity #3B.5 – Remote Management (Remote Wipe)*

Additional Information

A. Trigger Remote Management
  - The Issuer and the Remote-SE System can define some rules to automate the remote management of Mobile Payment Application in order to provision Payment Token (Single Use Key) or proactively manage the replacement of Card Profile when some conditions are satisfied (e.g. *Use of Virtual PAN*).
  - The remote management can also cover the monitoring of any Mobile Payment Application deployed in the field in order to collect status information and compare that information with the expected status of the Mobile Payment Application (e.g. number of Single Use Keys)
  - The Issuer may also decide to trigger some remote management of the Mobile Payment Application following a change of the Online PIN (when the Mobile PIN and the Online PIN share the same value) or when all the credentials pushed to a Mobile Payment Application must be wiped (e.g. termination of service or detection of misuse of the solution).
  - The Remote-SE System has a means to establish the link between a User identifier (*User_ID*) and the identifier (*RemSE_ID*) of a Remote-SE User Profile.

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 115

B. Session identifier

- ◊ Any remote notification message is associated with a session identifier (*Session_ID*) managed by the Remote-SE System.
- ◊ The session identifier will be used to build an Authentication Code and to derive a session key (*MS_Key*) from the Mobile Key (*M_Key*) shared between the Mobile Payment Application and the Remote-SE System.

C. Remote Notification message

- ◊ Depending on the remote management operation (*RemMgt_Info*) to be executed, the Remote-SE System builds the message to be notified to the Mobile Payment Application:
  - When preparing the delivery of a Card Profile (*PTP_CP*) using *Activity #3B.1 – Remote Management (PTP_CP)*, the Payment Credentials Management will have to construct the Payment Token Payload (PTP_CP) to be provisioned.
    At this stage of the process no data specific to the Card Profile has to be prepared.
  - When preparing the delivery of a Single Use Key (*PTP_SUK*) using *Activity #3B.2 – Remote Management (PTP_SUK)*, the Payment Credentials Management constructs the Payment Token Payload (PTP_SUK) to be provisioned.
    This operation requires a Card Profile to be available in the Mobile Payment Application.
    Moreover, the Remote-SE System must generate a random encryption key (*E_Key*) that is used to encrypt a message containing the *PTP_SUK*. The result of that encryption (*EPTP_SUK*) is added to the remote notification message while the encryption key is stored in the Remote-SE User Profile.
  - The notification to the Mobile Payment Application that the Mobile PIN has been changed (and all the stored PTP_SUK have to be removed) is done using *Activity #3B.3 – Remote Management (Mobile_PIN)* and does not require additional data.
  - The check of the status of the Mobile Payment Application is done using *Activity #3B.4 – Remote Management (Mobile Check)* and does not require additional data.
  - The notification to the Mobile Payment Application that its content must be wiped uses *Activity #3B.5 – Remote Management (Remote Wipe)* and does not require additional data.
- ◊ The notification message contains an identifier of the remote management operation to be executed (*RemMgt_Info*), the session identifier (*Session_ID*) and the value (*EPTP_SUK*) when a Single Use Key has to be provisioned to the Mobile Payment Application.
- ◊ The notification message is encrypted using the Mobile Key (*M_Key*) shared between the Remote-SE System and the Mobile Payment Application
- ◊ The content of the Remote-SE User Profile is updated with the session identifier (*Session_ID*) and status information.

D. Remote Notification delivery

- ◊ The Remote-SE System uses a Remote Notification Service to deliver the notification message to the Mobile Payment Application running on a given Mobile Device.
- ◊ Standard functions are used to support this process. From a functional point of view we consider that the Mobile Payment Application is identified using *Mobile_ID*. The actual mechanisms used to support this process are dependent of the Mobile Platform to be addressed.

E. Mobile Payment Application activation

- ◊ The Mobile Payment Application does not need to be running to receive a remote notification from the Remote-SE System.
- ◊ It is mandatory that the Mobile Payment Application is allowed to receive remote notification message. This access is granted either during the installation of the Mobile Payment Application or using an explicit acknowledgement from the User. It is not required to reiterate that acknowledgment for each notification message.
- ◊ The exact process when receiving a notification message and the interaction with the User is dependent of the Mobile Platform. The Mobile Payment Application is started either using an automated process upon reception of the remote notification or when the User requests the notification to be processed.

F. Process Remote Notification

- The Mobile Payment Application must always check its integrity at startup and perform some rooted device detection.
- *The security model assumes that there is no 100% guarantee that the Mobile Payment Application is able to detect a rooted device or detect that the device had been previously rooted and was "unrooted" afterwards.*
- The Mobile Payment Application regenerates the Storage Key (*ST_Key*) to access the local encrypted database.
- The Mobile Payment Application retrieves the Mobile Key (*M_Key*) from the local encrypted database.
- That key is used to decrypt the remote notification message. The length value stored in the message is compared with the actual length of the decrypted message.
- Upon successful validation, the Mobile Payment Application retrieves the remote management operation (*RemMgt_Info*) to be executed.

G. Connection to Remote-SE System

- The Mobile Payment Application retrieves a URL (*Notification_URL*) from the local encrypted database.
- The Mobile Payment Application connects to the Remote-SE System using that URL.
- The mobile application validates the SSL/TLS connection (Certificates) and checks the Common Name (CN) against a value[8] stored in the local encrypted database.
  At this stage the server authentication process is completed.
- The Mobile Payment Application uses its identifier (*RemSE_ID*) stored in the local encrypted database and the session identifier (*Session_ID*) retrieved from the decrypted notification message to build an Authentication Code (*Authentication_Code*).
- The Mobile Payment Application sends the Authentication Credentials to the Remote-SE System for validation.

H. Authentication Code validation

- The Remote-SE System identifies the Remote-SE User Profile using the identifier (*RemSE_ID*) provided by the Mobile Payment Application.
- The Remote-SE System retrieves the session identifier from that profile and generates an Authentication Code.
- The generated Authentication Code is checked against the received value to complete the client authentication process.
- At this stage the mutual authentication process is completed and the processing of the remote management operation (*RemMgt_Info*) can start using one of the following activities:
  - *Activity #3B.1 – Remote Management (PTP_CP)*
  - *Activity #3B.2 – Remote Management (PTP_SUK)*
  - *Activity #3B.3 – Remote Management (Mobile_PIN)*
  - *Activity #3B.4 – Remote Management (Mobile Check)*
  - *Activity #3B.5 – Remote Management (Remote Wipe)*

[8] That value is also part of the object (*Notification_URL*).

IPR2025-01152
Apple EX1008 Page 117

Activity #3B.1 – Remote Management (PTP_CP)

The delivery of Card Profile (*PTP_CP*) using remote management is performed using first a trigger process described in *Activity #3A – Remote Management (Trigger)* and then executing the process defined in this activity.
This section provides first an *Overview* of the activity (PTP_CP remote management) and then provides *Additional Information*.

The detailed description of the flow used to support *Activity #3B.1 – Remote Management (PTP_CP)* is provided in the *Appendix A – Technical Description*.

Overview

The ***Remote Management (Delivery of Card Profile)*** activity uses the following components:

- User
- Mobile Device
- Mobile Payment Application
- Remote-SE System (Payment Credentials Management)
- Remote-SE System (Remote Notification Service)
- Issuer

The process supported by this activity aims to:

A. Build the Payment Token Payload (Card Profile),
B. Encrypt the data for delivery to the Mobile Payment Application,
C. Deliver the encrypted *PTP_CP*,
D. Generate Mobile Session Key (*MS_Key*) and decrypt message from Remote-SE System,
E. Validate message content and generate an Activation Proof (*PTP_ActProof*),
F. Store PTP_CP and remove any available *PTP_SUK* from local encrypted database,
G. Deliver the Activation Proof to Remote-SE System,
H. Validate Activation Proof and update Remote-SE User Profile,
I. Inform Issuer and complete process with Mobile Payment Application,
J. Complete process and wipe any temporary value.
K. Support Fraud Management.

At the end of the process, the status is the following:

- The Mobile Payment Application (on the Mobile Device) has stored a Card Profile (*PTP_CP*) in the local encrypted database.
- The Mobile Payment Application is ready to be provisioned with at least one Payment Token (Single Use Key) using *Activity #3B.2 – Remote Management (PTP_SUK)*.
- The Remote-SE System drives the remote management of the Mobile Payment Application.

Additional Information

A. Payment Token Payload (Card Profile)
  - The Payment Credentials Management collects the payment credentials to be used.
  - The Payment Credentials Management formats the information to build the Card Profile (*PTP_CP*) and store the information in the Remote-SE User Profile.

B. Encryption process
  - The Remote-SE System builds a message containing the *PTP_CP*.
  - The Remote-SE System generates a Mobile Session Key (*MS_Key*) using the Mobile Key (*M_Key*) and the session identifier (*Session_ID*) defined for this remote management operation.
  - The Remote-SE System encrypts the message using the Mobile Session Key (*MS_Key*)

C. PTP_CP delivery
  - The Remote-SE System delivers the encrypted message to the Mobile Payment Application using the SSL/TLS channel established during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*).

D. Decryption process
- ◊ The Mobile Payment Application has already defined the Mobile Session Key (*MS_Key*) during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*)
- ◊ That key is used to decrypt the message from the Remote-SE System. The length value stored in the message is compared with the actual length of the decrypted message.

E. Validation and Activation
- ◊ Upon successful validation, the Mobile Payment Application retrieves the Card Profile (*PTP_CP*).
- ◊ The Mobile Payment Application prepares a proof for activation of the received Payment Credentials (*PTP_ActProof*).
  The Activation Proof contains information about the previous Card Profile (if any), the new Card Profile and all the stored Single Use Keys (if any).

F. Payment Credentials Management
- ◊ The Mobile Payment Application removes the previous Card Profile (if any) and stores in the local encrypted database the received Card Profile.
- ◊ The Mobile Payment Application removes any stored Single Use Key.
- ◊ The Mobile Payment Application updates the status and counter stored in the local encrypted database.

G. Activation Proof delivery
- ◊ The Mobile Payment Application prepares a message containing the Activation Proof.
- ◊ The Mobile Payment Application encrypts this message using the Mobile Session Key (*MS_Key*).
- ◊ The Mobile Payment Application sends the encrypted message to the Remote-SE System for validation.

H. Activation Proof validation
- ◊ The Remote-SE System decrypts the message using the Mobile Session Key (*MS_Key*).
- ◊ The length value stored in the message is compared with the actual length of the decrypted message.
- ◊ The Remote-SE System validates the content of the Activation Proof using the data stored in the Remote-SE User Profile

I. Complete process (Remote-SE System)
- ◊ The Remote-SE System notifies the Issuer about the outcome of the provisioning of a Card Profile to the Mobile Payment Application.
- ◊ The Remote-SE System updates the status and counter contained in the Remote-SE User Profile.
- ◊ The Remote-SE System delivers a return code to the Mobile Payment Application to notify the completion of the process

J. Complete process (Mobile Payment Application)
- ◊ The Mobile Payment Application analyzes the return code from the Remote-SE System and completes the update of the status stored in the local encrypted database.
- ◊ The Mobile Payment Application wipes any temporary value used during the process such as the Mobile Session Key (*MS_Key*) and the Storage Key (*ST_Key*).

K. Fraud Management
- ◊ A Card Profile (including sensitive Payment Credentials) has been delivered to the Mobile Payment Application.
- ◊ No *PayPass* transaction is allowed at this stage for the selected product.

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 119

Activity #3B.2 – Remote Management (PTP_SUK)

The delivery of Single Use Key (*PTP_SUK*) using remote management is performed using first a trigger process described in *Activity #3A – Remote Management (Trigger)* and then executing the process defined in this activity.
This section provides first an *Overview* of the activity (PTP_SUK remote management) and then provides *Additional Information*.

The detailed description of the flow used to support *Activity #3B.2 – Remote Management (PTP_SUK)* is provided in the *Appendix A – Technical Description*.

Overview

The ***Remote Management (Delivery of Single Use Key)*** activity uses the following components:

- User
- Mobile Device
- Mobile Payment Application
- Remote-SE System (Payment Credentials Management)
- Remote-SE System (Remote Notification Service)
- Issuer

The process supported by this activity aims to:

A. Retrieve the encryption key (*E_Key*) from the Remote-SE User Profile,
B. Encrypt the data for delivery to the Mobile Payment Application,
C. Deliver the encrypted *E_Key*,
D. Generate Mobile Session Key (*MS_Key*) and decrypt message from Remote-SE System,
E. Validate message content and store *PTP_SUK* in local encrypted database,
F. Generate an Activation Proof (*PTP_ActProof*),
G. Deliver the Activation Proof to Remote-SE System,
H. Validate Activation Proof and update Remote-SE User Profile,
I. Optionally inform Issuer and complete process with Mobile Payment Application,
J. Complete process and wipe any temporary value.
K. Support Fraud Management

At the end of the process, the status is the following:

- The Mobile Payment Application (on the Mobile Device) has stored a Card Profile (*PTP_CP*) and at least one Single Use Key (*PTP_SUK*) in the local encrypted database.
- The Mobile Payment Application is ready to:
  - Be provisioned with additional Payment Token (Single Use Key) using *Activity #3B.2 – Remote Management (PTP_SUK)*.
  - Perform a Mobile *PayPass* transaction using *Activity #4 – Payment (PayPass transaction)*.
- The Remote-SE System drives the remote management of the Mobile Payment Application.

Additional Information

A. Encryption key
  - The Remote-SE System (Payment Credentials Management) retrieves the encryption key (*E_Key*) from the Remote-SE User Profile.
  - This encryption key was used to encrypt the *PTP_SUK* and deliver *EPTP_SUK* to the Mobile Payment Application using a remote notification message during *Activity #3A – Remote Management (Trigger)*.

B. Encryption process
  - The Remote-SE System builds a message containing the *E_Key*.
  - The Remote-SE System generates a Mobile Session Key (*MS_Key*) using the Mobile Key (*M_Key*) and the session identifier (*Session_ID*) defined for this remote management operation.
  - The Remote-SE System encrypts the message using the Mobile Session Key (*MS_Key*)

C. Encryption Key delivery
  - The Remote-SE System delivers the encrypted message to the Mobile Payment Application using the SSL/TLS channel established during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*).

IPR2025-01152
Apple EX1008 Page 120

D. Decryption process
- ◊ The Mobile Payment Application has already defined the Mobile Session Key (*MS_Key*) during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*)
- ◊ That key is used to decrypt the message from the Remote-SE System. The length value stored in the message is compared with the actual length of the decrypted message.

E. Validation and Storage
- ◊ Upon successful validation, the Mobile Payment Application retrieves the encryption key (*E_Key*).
- ◊ That key is used to decrypt the encrypted *PTP_SUK* delivered using *EPTP_SUK* value available in the remote notification message sent by the Remote-SE System.
  The length value stored in the message (i.e. decrypted *EPTP_SUK*) is compared with the actual length of the decrypted message.
- ◊ The Mobile Payment Application has now access to the Single Use Key (*PTP_SUK*).
- ◊ The Mobile Payment Application stores *PTP_SUK* in the local encrypted database and updates status information.

F. Activation Proof generation
- ◊ The Mobile Payment Application prepares a proof for activation of the received Payment Credentials (*PTP_ActProof*).
  The Activation Proof contains information about the Card Profile and all the stored Single Use Keys.

G. Activation Proof delivery
- ◊ The Mobile Payment Application prepares a message containing the Activation Proof.
- ◊ The Mobile Payment Application encrypts this message using the Mobile Session Key (*MS_Key*).
- ◊ The Mobile Payment Application sends the encrypted message to the Remote-SE System for validation.

H. Activation Proof validation
- ◊ The Remote-SE System decrypts the message using the Mobile Session Key (*MS_Key*).
- ◊ The length value stored in the message is compared with the actual length of the decrypted message.
- ◊ The Remote-SE System validates the content of the Activation Proof using the data stored in the Remote-SE User Profile

I. Complete process (Remote-SE System)
- ◊ The Remote-SE System optionally notifies the Issuer about the outcome of the provisioning of a Single Use Key to the Mobile Payment Application.
- ◊ The Remote-SE System updates the status and counter contained in the Remote-SE User Profile.
- ◊ The Remote-SE System delivers a return code to the Mobile Payment Application to notify the completion of the process

J. Complete process (Mobile Payment Application)
- ◊ The Mobile Payment Application analyzes the return code from the Remote-SE System and completes the update of the status and counter stored in the local encrypted database.
- ◊ The Mobile Payment Application wipes any temporary value used during the process such as the Mobile Session Key (*MS_Key*) and the Storage Key (*ST_Key*).

K. Fraud Management
- ◊ The Mobile Payment Application contains the Payment Credentials (*PTP_CP* and *PTP_SUK*) required to support a Mobile PayPass transaction.
- ◊ The Issuer allows Mobile *PayPass* transaction at this stage for the selected product.
- ◊ The Mobile PayPass transaction must use the credentials provisioned to the Mobile Payment Application with a set of controls defined at Issuer level (e.g. Use provisioned ATC, ATC replay detection).

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 121

Activity #3B.3 – Remote Management (Mobile_PIN)

The remote management of the Mobile Payment Application is performed using first a trigger process described in *Activity #3A – Remote Management (Trigger)* and then executing the process defined in this activity.
This section provides first an *Overview* of the activity (Mobile PIN remote management) and then provides *Additional Information*.

When the Mobile PIN is changed, all the stored[9] Single Use Keys (PTP_SUK) have to be removed from the local encrypted database used by the Mobile Payment Application. The status and counter have to be updated.

The detailed description of the flow used to support *Activity #3B.3 – Remote Management (Mobile_PIN)* is provided in the *Appendix A – Technical Description*.

Overview

The ***Remote Management (Mobile PIN)*** activity uses the following components:

- User
- Mobile Device
- Mobile Payment Application
- Remote-SE System (Payment Credentials Management)
- Remote-SE System (Remote Notification Service)
- Issuer

The process supported by this activity aims to:

A. Build a message with the confirmation of the remote management action,
B. Encrypt the message for delivery to the Mobile Payment Application,
C. Generate Mobile Session Key (*MS_Key*) and decrypt message from Remote-SE System,
D. Validate message content and execute the action: remove any stored PTP_SUK as result of change of Mobile PIN value.
E. Generate a status message reusing the concept of Activation Proof (*PTP_ActProof*).
There is no activation but we use that container to carry information about the Card Profile.
F. Deliver the Activation Proof to Remote-SE System,
G. Validate Activation Proof (= Status information) and update Remote-SE User Profile,
H. Optionally inform Issuer and complete process with Mobile Payment Application,
I. Complete process and wipe any temporary value.
J. Support Fraud Management

At the end of the process, the status is the following:

- The Mobile Payment Application (on the Mobile Device) has a Card Profile (*PTP_CP*) in the local encrypted database (if a Card Profile had been previously provisioned to the Mobile Payment Application) but the Mobile Payment Application does not have any Single Use Key available to support a Mobile *PayPass* transaction using *Activity #4 – Payment (PayPass transaction)*.
- The Mobile Payment Application is ready to be provisioned with at least one Payment Token (Single Use Key) using *Activity #3B.2 – Remote Management (PTP_SUK)*.
- The Remote-SE System drives the remote management of the Mobile Payment Application.

Additional Information

A. Remote Management
  - The Remote-SE System (Payment Credentials Management) builds a message using *RemMgt_Info* to confirm the operation initially defined in the remote notification message.

[9] Those keys were protected using the previous value of the Mobile PIN before Online PIN change (and therefore Mobile PIN change when Online PIN and Mobile PIN are synchronized).

B. Delivery process

- The Remote-SE System generates a Mobile Session Key (*MS_Key*) using the Mobile Key (*M_Key*) and the session identifier (*Session_ID*) defined for this remote management operation.
- The Remote-SE System encrypts the message using the Mobile Session Key (*MS_Key*)
- The Remote-SE System delivers the encrypted message to the Mobile Payment Application using the SSL/TLS channel established during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*).

C. Decryption process

- The Mobile Payment Application has already defined the Mobile Session Key (*MS_Key*) during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*)
- That key is used to decrypt the message from the Remote-SE System. The length value stored in the message is compared with the actual length of the decrypted message.

D. Validation and Execution

- Upon successful validation, the Mobile Payment Application executes the process associated with that remote management operation.
- The Mobile Payment Application removes any *PTP_SUK* stored the local encrypted database.
- The Mobile Payment Application updates status and counter information.

E. Activation Proof generation

- The Mobile Payment Application prepares a confirmation message for the Remote-SE System.
- The process used for confirmation reuses the concept of Activation Proof (*PTP_ActProof*). It contains information about the Card Profile.

F. Activation Proof delivery

- The Mobile Payment Application prepares a message containing the Activation Proof.
- The Mobile Payment Application encrypts this message using the Mobile Session Key (*MS_Key*).
- The Mobile Payment Application sends the encrypted message to the Remote-SE System for validation.

G. Activation Proof validation

- The Remote-SE System decrypts the message using the Mobile Session Key (*MS_Key*).
- The length value stored in the message is compared with the actual length of the decrypted message.
- The Remote-SE System validates the content of the Activation Proof using the data stored in the Remote-SE User Profile.

H. Complete process (Remote-SE System)

- The Remote-SE System optionally notifies the Issuer about the outcome of the execution of the remote management action.
- The Remote-SE System updates the status and counter contained in the Remote-SE User Profile.
- The Remote-SE System delivers a return code to the Mobile Payment Application to notify the completion of the process.

I. Complete process (Mobile Payment Application)

- The Mobile Payment Application analyzes the return code from the Remote-SE System and can optionally inform the User.
- The Mobile Payment Application wipes any temporary value used during the process such as the Mobile Session Key (*MS_Key*) and the Storage Key (*ST_Key*).

J. Fraud Management

- A Card Profile was previously delivered to the Mobile Payment Application.
- No *PayPass* transaction is allowed at this stage for the selected product (All the Single Use Keys have been wiped and are *de facto* revoked at Issuer level).

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 123

Activity #3B.4 – Remote Management (Mobile Check)

The remote management of the Mobile Payment Application is performed using first a trigger process described in *Activity #3A – Remote Management (Trigger)* and then executing the process defined in this activity.
This section provides first an *Overview* of the activity (remote mobile check)and then provides *Additional Information.*

This activity is used to retrieve detailed status of the Mobile Payment Application including information about the Card Profile (if any) and the Single Use Keys (if any).

The detailed description of the flow used to support *Activity #3B.4 – Remote Management (Mobile Check)* is provided in the *Appendix A – Technical Description.*

Overview

The ***Remote Management (Mobile Check)*** activity uses the following components:

- User
- Mobile Device
- Mobile Payment Application
- Remote-SE System (Payment Credentials Management)
- Remote-SE System (Remote Notification Service)
- Issuer

The process supported by this activity aims to:

A. Build a message with the confirmation of the remote management action,
B. Encrypt the message for delivery to the Mobile Payment Application,
C. Generate Mobile Session Key (*MS_Key*) and decrypt message from Remote-SE System,
D. Validate message content and execute the action: retrieve status information from the Mobile Payment Application.
E. Generate a status message reusing the concept of Activation Proof (*PTP_ActProof*).
There is no activation but we use that container to carry status information about the Card Profile (if any) and Single Use Keys (if any).
F. Deliver the Activation Proof to Remote-SE System,
G. Use Activation Proof (= Status information) to update Remote-SE User Profile,
H. Optionally inform Issuer and complete process with Mobile Payment Application,
I. Complete process and wipe any temporary value.
J. Optionally support Fraud Management

At the end of the process, the status is the following:

- The status of the Mobile Payment Application (on the Mobile Device) remains unchanged.
- The Remote-SE System drives the remote management of the Mobile Payment Application.

Additional Information

A. Remote Management
  - The Remote-SE System (Payment Credentials Management) builds a message using *RemMgt_Info* to confirm the operation initially defined in the remote notification message.

B. Delivery process
  - The Remote-SE System generates a Mobile Session Key (*MS_Key*) using the Mobile Key (*M_Key*) and the session identifier (*Session_ID*) defined for this remote management operation.
  - The Remote-SE System encrypts the message using the Mobile Session Key (*MS_Key*)
  - The Remote-SE System delivers the encrypted message to the Mobile Payment Application using the SSL/TLS channel established during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*).

C. Decryption process
  - The Mobile Payment Application has already defined the Mobile Session Key (*MS_Key*) during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*)
  - That key is used to decrypt the message from the Remote-SE System. The length value stored in the message is compared with the actual length of the decrypted message.

D. Validation and Execution
- ◊ Upon successful validation, the Mobile Payment Application executes the process associated with that remote management operation.
- ◊ The Mobile Payment Application removes any *PTP_SUK* stored the local encrypted database.
- ◊ The Mobile Payment Application updates status and counter information.

E. Activation Proof generation
- ◊ The Mobile Payment Application prepares a status message for the Remote-SE System.
- ◊ The process used for status reuses the concept of Activation Proof (*PTP_ActProof*). It contains information about the Card Profile and all the stored Single Use Keys.

F. Activation Proof delivery
- ◊ The Mobile Payment Application prepares a message containing the Activation Proof.
- ◊ The Mobile Payment Application encrypts this message using the Mobile Session Key (*MS_Key*).
- ◊ The Mobile Payment Application sends the encrypted message to the Remote-SE System.

G. Use Activation Proof
- ◊ The Remote-SE System decrypts the message using the Mobile Session Key (*MS_Key*).
- ◊ The length value stored in the message is compared with the actual length of the decrypted message.
- ◊ The Remote-SE System can use the content of the Activation Proof to check (or update) data stored in the Remote-SE User Profile.
- ◊ The Remote-SE System can inform the Issuer if there is any discrepancy between the expected status of the Mobile Payment Application (as reported in the Remote-SE User Profile) compared to what has been reported by the Mobile Payment Application.

H. Complete process (Remote-SE System)
- ◊ The Remote-SE System optionally notifies the Issuer about the outcome of the execution of the remote management action.
- ◊ The Remote-SE System may update the status and counter contained in the Remote-SE User Profile.
- ◊ The Remote-SE System delivers a return code to the Mobile Payment Application to notify the completion of the process.

I. Complete process (Mobile Payment Application)
- ◊ The Mobile Payment Application analyzes the return code from the Remote-SE System and might inform the User.
- ◊ The Mobile Payment Application wipes any temporary value used during the process such as the Mobile Session Key (*MS_Key*) and the Storage Key (*ST_Key*).

J. Fraud Management
- ◊ There is an opportunity for the Issuer to validate stored information (used to support Fraud Management) against the current status of the Mobile Payment Application.

Activity #3B.5 – Remote Management (Remote Wipe)

The remote management of the Mobile Payment Application is performed using first a trigger process described in *Activity #3A – Remote Management (Trigger)* and then executing the process defined in this activity.
This section provides first an *Overview* of the activity (remote wipe) and then provides *Additional Information*.

This activity is used to wipe the content of the Mobile Payment Application and remove any stored credentials (i.e. Card Profile (if any) and Single Use Keys (if any)) and data used to support the Mobile Payment Application (Local encrypted database, parameters…).

The detailed description of the flow used to support *Activity #3B.5 – Remote Management (Remote Wipe)* is provided in the *Appendix A – Technical Description*.

Overview

The ***Remote Management (Remote Wipe)*** activity uses the following components:

- User
- Mobile Device
- Mobile Payment Application
- Remote-SE System (Payment Credentials Management)
- Remote-SE System (Remote Notification Service)
- Issuer

The process supported by this activity aims to:

A. Build a message with the confirmation of the remote management action,
B. Encrypt the message for delivery to the Mobile Payment Application,
C. Generate Mobile Session Key (*MS_Key*) and decrypt message from Remote-SE System,
D. Validate message content and execute the action:
  - Generate a status message reusing the concept of Activation Proof (*PTP_ActProof*). There is no activation but we use that container to carry information about the Card Profile (if any) and Single Use Keys (if any) before removal.
  - Wipe any stored credentials (Card Profile and Single Use Keys), the content of the local encrypted database, the database itself and any stored information.

E. Deliver the Activation Proof to Remote-SE System,
F. Use Activation Proof (= Status information) and inform Issuer,
G. Complete process,
H. Inform the User.
I. Support Fraud Management.

At the end of the process, the status is the following:

- The Mobile Payment Application (on the Mobile Device) does not have any credentials left and must be reinitialized using *Activity #2 – Initialization*.
- The initialization process requires an Activation Code (*Activation_Code*).
- The User can use the *Activity #1 – Registration & Installation* to receive that Activation Code but there is no need to reinstall the Mobile Payment Application.

Additional Information

A. Remote Management
  - The Remote-SE System (Payment Credentials Management) builds a message using *RemMgt_Info* to confirm the operation initially defined in the remote notification message.

B. Delivery process
  - The Remote-SE System generates a Mobile Session Key (*MS_Key*) using the Mobile Key (*M_Key*) and the session identifier (*Session_ID*) defined for this remote management operation.
  - The Remote-SE System encrypts the message using the Mobile Session Key (*MS_Key*)
  - The Remote-SE System delivers the encrypted message to the Mobile Payment Application using the SSL/TLS channel established during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*).

IPR2025-01152
Apple EX1008 Page 126

C. Decryption process

- The Mobile Payment Application has already defined the Mobile Session Key (*MS_Key*) during the first part of the remote management operation (*Activity #3A – Remote Management (Trigger)*).
- That key is used to decrypt the message from the Remote-SE System. The length value stored in the message is compared with the actual length of the decrypted message.

D. Validation and Execution

- Upon successful validation, the Mobile Payment Application executes the process associated with that remote management operation.
- The Mobile Payment Application prepares a status message for the Remote-SE System.
- The process used for status reuses the concept of Activation Proof (*PTP_ActProof*). It contains information about the Card Profile (if any) and the Single Use Keys (if any).
- The Mobile Payment Application prepares a message containing the Activation Proof.
- The Mobile Payment Application encrypts this message using the Mobile Session Key (*MS_Key*).
- The Mobile Payment Application wipes any stored credentials (Card Profile and Single Use Keys), the content of the local encrypted database, the database itself and any stored information.

E. Activation Proof delivery

- The Mobile Payment Application sends the encrypted message to the Remote-SE System.

F. Use Activation Proof

- The Remote-SE System decrypts the message using the Mobile Session Key (*MS_Key*).
- The length value stored in the message is compared with the actual length of the decrypted message.
- The Remote-SE System can use the content of the Activation Proof to complete the removal of the Remote-SE User Profile.
- The Remote-SE System must inform the Issuer that the Card Profile and all related Single Use Keys are not anymore valid.

G. Complete process

- The Remote-SE System completes the process and informs the Mobile Payment Application.
- The Mobile Payment Application must wipe any residual data and is reset to 'not initialized'.

H. Inform User

- The Mobile Payment Application must inform the User that the Mobile Payment Application cannot be used anymore without a new initialization.

I. Fraud Management

- The User is still registered by the content of the Mobile Payment Application has been wiped.
- No *PayPass* transaction is allowed at this stage for the selected product.
  The Card Profile and all the Single Use Keys (if any) have been wiped.
  Those payment credentials are *de facto* revoked at Issuer level.

Activity #4 – Payment (*PayPass* transaction)

This section provides first an *Overview* of the activity and then provides *Additional Information*.

The detailed description of the flow used to support *Activity #4 – Payment (PayPass transaction)* is provided in the *Appendix A – Technical Description*.

Overview

The ***Payment (PayPass transaction)*** activity uses the following components:

- User
- Mobile Device
- Mobile Payment Application
- Remote-SE System (Payment Credentials Management)
- Merchant (*PayPass* POS)
- Acquirer
- Payment Network
- Issuer

The process supported by this activity aims to:

A. Start the Mobile Payment Application and validate its integrity,
B. Get access to the local encrypted storage,
C. Check if a Card Profile is available with at least one Single Use Key,
D. Execute the Remote-SE Payment Transaction Flow,
E. Use the standard process for online authorization of a *PayPass* transaction,
F. Optionally synchronize information between the Issuer and the Remote-SE System,
G. Complete process, inform User and wipe any temporary value.
H. Support Fraud Management.

At the end of the process, the status is the following:

- The Mobile Payment Application (on the Mobile Device) has a Card Profile (*PTP_CP*) in the local encrypted database (if a Card Profile had been previously provisioned to the Mobile Payment Application) and we have one of the following cases:
  - The Mobile Payment Application does not have anymore any Single Use Key (*PTP_SUK*) available to support a Mobile *PayPass* transaction.
    The Mobile Payment Application must be provisioned with at least one Payment Token (Single Use Key) using *Activity #3B.2 – Remote Management (PTP_SUK)* to get access to the *Activity #4 – Payment (PayPass transaction)*.
  - The Mobile Payment Application has at least one Single Use Key available to support a Mobile *PayPass* transaction using *Activity #4 – Payment (PayPass transaction)*.
    The Mobile Payment Application is ready to be provisioned with additional Payment Token (Single Use Key) using *Activity #3B.2 – Remote Management (PTP_SUK)*.
- The Remote-SE System drives the remote management of the Mobile Payment Application.

Additional Information

A. Application start
  - The Mobile Payment Application performs some integrity checks at startup and detection for rooted device.
  - *The security model assumes that there is no 100% guarantee that the Mobile Payment Application is able to detect a rooted device or detect that the device had been previously rooted and was "unrooted" afterwards.*

B. Local Encrypted Database
  - The Mobile Payment Application regenerates the Storage Key (*ST_Key*) to access the local encrypted database.

C. Payment Credentials Availability
  - The Mobile Payment Application checks the availability and status of Payment Credentials in the local encrypted database using the following information:
    - Payment credentials (*PTP_CP* and list of *PTP_SUK*)
    - Status and counter information (*PTPCP_Status*, *PTPSUK_Status* and *PTPSUK_Counter*)

IPR2025-01152
Apple EX1008 Page 128

- ◊ The Mobile Payment Application informs the User and can support the *PayPass* transaction if at least one Card Profile (*PTP_CP*) and one Single Use Key (*PTP_SUK*) are available.

D. Remote-SE Payment Transaction Flow

- ◊ The Mobile Payment Application uses the flow defined in section *PayPass Transaction* of *Appendix B – Payment Transaction Flow*.

E. Online authorization of *PayPass* transaction

- ◊ The *PayPass* transaction is authorized online.
- ◊ The Issuer identifies (e.g. using PAN) a transaction performed using Remote-SE Mobile *PayPass* solution
- ◊ The Issuer validates the cryptogram (AC or CVC3) generated during the *PayPass* transaction.
  - ▪ When processing a *PayPass* M/Chip transaction the standard process applies.
  - ▪ When processing a *PayPass* Mag Stripe transaction, the Issuer must add an additional step in the CVC3 validation in order to consider the additional session key derivation which was used by the Payment Credentials Management when generating the Single Use Key delivered to the Mobile Payment Application.
- ◊ When the cryptogram validation fails, the Issuer must consider that the error may come from the use of a wrong Mobile PIN value. The Issuer must update the fraud detection status mechanisms (e.g. Mobile PIN Try Counter) accordingly.
- ◊ The same way, when a cryptogram validation is successful, the Issuer must integrate this event in the fraud detection and monitoring tools.
- ◊ The Issuer can also consider additional parameters (e.g. Application Transaction Counter) for monitoring and detection of any misuse of the Remote-SE Mobile *PayPass*.

F. Synchronization

- ◊ The Issuer can synchronize information with the Remote-SE System in order to update or feed the rules used to trigger the delivery of additional Payment Tokens (Single Use Keys) or any remote management operation.

G. Complete process (Mobile Payment Application)

- ◊ The Mobile Payment Application wipes any temporary value used during the process such as the Storage Key (*ST_Key*) or the decrypted Payment Credentials.
- ◊ The Mobile Payment Application must inform the User about the outcome of the *PayPass* transaction.
- ◊ The Mobile Payment Application must inform the User about the availability of Payment Tokens (Single Use Keys) which can be used to support at least one new Mobile *PayPass* transaction.

H. Fraud Management

- ◊ The Mobile PayPass transaction must use the credentials provisioned to the Mobile Payment Application with a set of controls defined at Issuer level (e.g. Use provisioned ATC, ATC replay detection).
- ◊ A Card Profile was previously delivered to the Mobile Payment Application.
- ◊ If no more Single Use Key is available (in the Mobile Payment Application) then no *PayPass* transaction is allowed at this stage for the selected product.
- ◊ Otherwise, the Mobile Payment Application contains the Payment Credentials (*PTP_CP* and *PTP_SUK*) required to support a Mobile PayPass transaction. The Issuer allows Mobile *PayPass* transaction at this stage for the selected product.

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 129

Appendix A – Technical Description

Overview

This appendix contains the following sections:

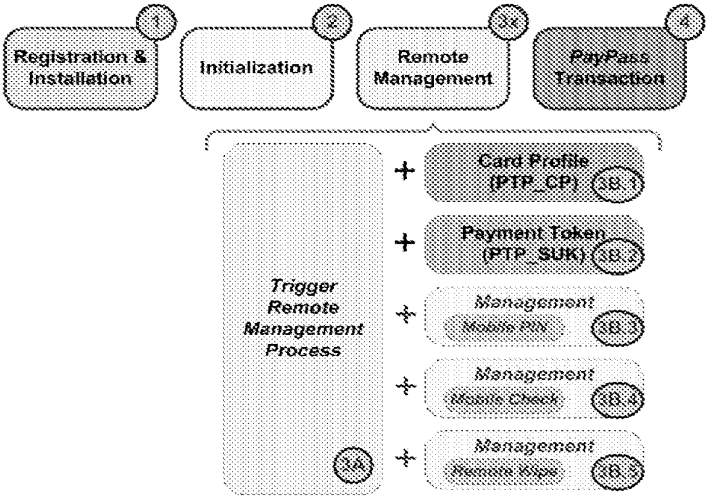
- *Detailed Activity Flows*
  This section contains the detailed description of the activity flows used to support Remote-SE Mobile *PayPass.*
- *Key Management*
  This section explains the management of the keys used to support Remote-SE Mobile *PayPass.*
- *Functions*
  This section lists the functions used to support Remote-SE Mobile *PayPass.*
- *Payment Functions*
  This section lists the functions used to support payment related processes of Remote-SE Mobile *PayPass.*
- *Data Elements*
  This section lists the data elements used to support Remote-SE Mobile *PayPass.*

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 130

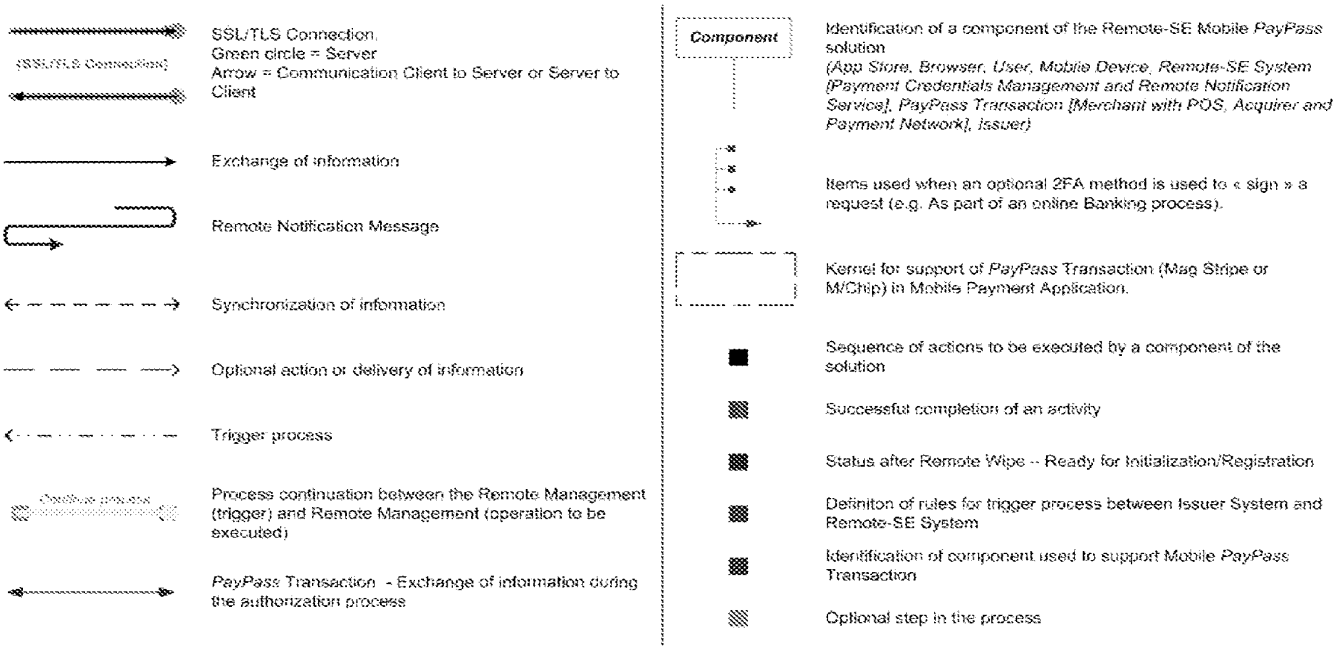
Detailed Activity Flows

This section contains the detailed description of the activity flows used to support Remote-SE Mobile *PayPass*:



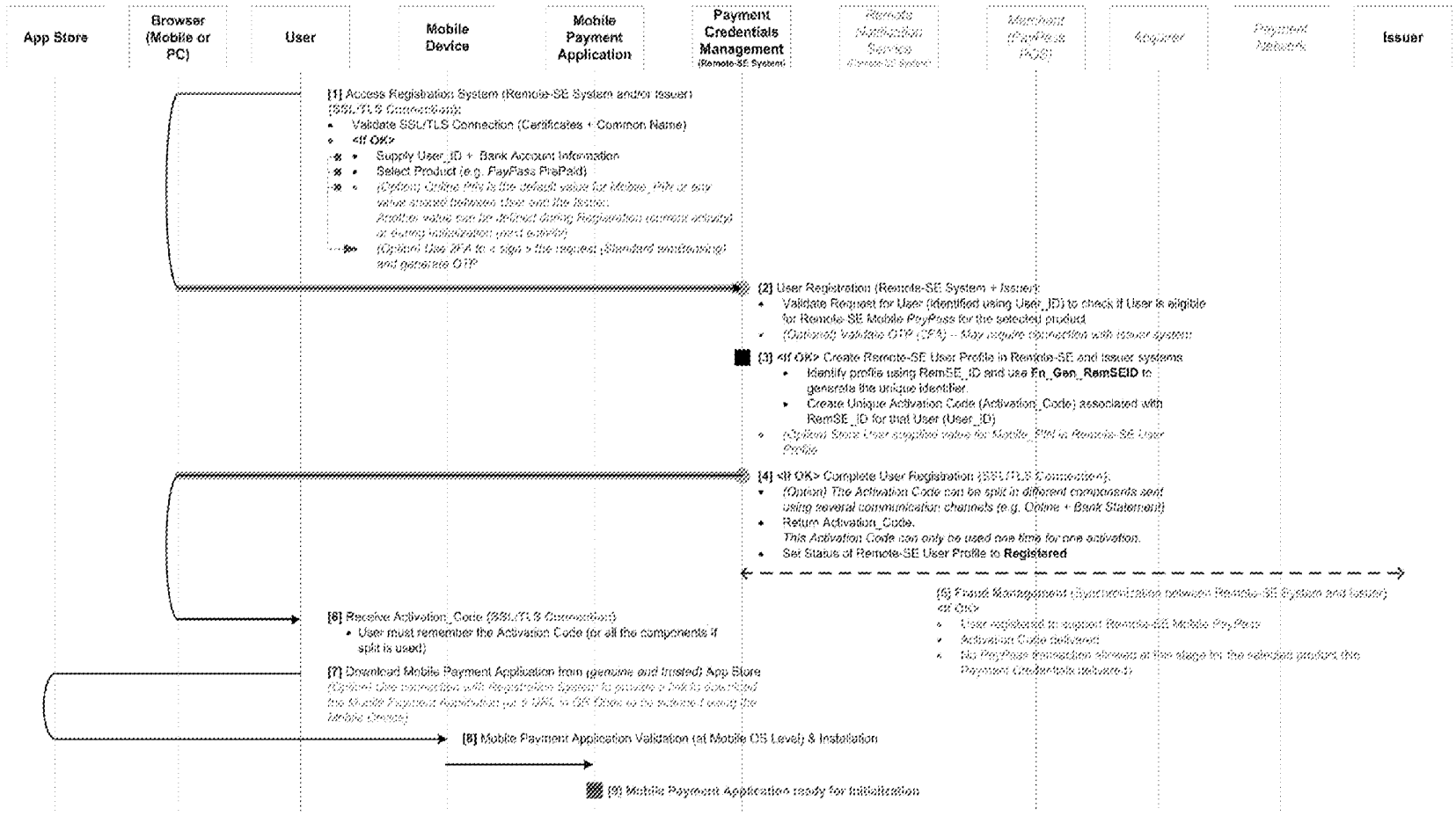
- *Activity #1 – Registration & Installation*
- *Activity #2 – Initialization*
- *Activity #3 – Remote Management*
  - *Activity #3A – Remote Management (Trigger)*
  - *Activity #3B.1 – Remote Management (PTP_CP)*
  - *Activity #3B.2 – Remote Management (PTP_SUK)*
  - *Activity #3B.3 – Remote Management (Mobile_PIN)*
  - *Activity #3B.4 – Remote Management (Mobile Check)*
  - *Activity #3B.5 – Remote Management (Remote Wipe)*
- *Activity #4 – Payment (PayPass transaction)*

The following symbols / notations are used in the activity flows presented in this section:



IPR2025-01152
Apple EX1008 Page 131

Activity #1 – Registration & Installation



Activity #2 – Initialization

App Store | Browser (Mobile or PC) | User | Mobile Device | Mobile Payment Application | Payment Credentials Management (Remote-SE System) | Remote Notification Service (Remote-SE System) | Merchant (Mobile or PC) | Acquirer | Payment Network | Issuer

[1] Start Mobile Payment Application (1st Use)

[2] Integrity Check / Rooted Device Check
<if NOK> UI to Inform User
Ask User Credentials:
- User_ID
- Activation_Code *(Provided during Registration Process)*
- *[Option] Online PIN is the default value for Mobile_PIN or any value chosen between User and the Issuer.*
  Another value can be defined during Registration (previous activity) or during Initialization (current activity)
- *(Option) Use 2FA to sign + the request (Standard eBanking) and generate OTP*

[3] Mobile Payment Application to generate
- Mobile_ID (Unique ID) used for Remote Notification to Mobile Payment Application
- Generate Random Value (Rnd_Storage) and store value in Mobile Payment Application Data
- Generate Mobile Storage Key (ST_Key) using **Fn_Gen_STKey:**
  - Information about Mobile Device
  - Information embedded in Mobile Payment Application (Code)
  - Random value generated by Mobile Payment Application (Rnd_Storage)

  This key is used for encrypted local storage of data

[4] Connect to Remote-SE System using Initialization_URL defined in Mobile Payment Application *(SSL/TLS Connection)*:
(Info is either hardcoded (default) or the issuer stage provide it as part of the Registration activity – e.g. QR Code)
- Validate SSL/TLS Connection (Certificates + Common Name)
- <if OK> Send Initialization Credentials
- User_ID + Activation_Code
  (Can only be used one time)
- Mobile_ID
- *[Option] Mobile PIN*
- *[Option] When 2FA is used, provide OTP*
- *Wipe Mobile Pin value from Memory (in defined at this stage)*

[5] Remote-SE System to:
- Retrieve RemSE_ID value using **Fn_Get_RemSEID** with User_ID to access Remote-SE User Profile (RemSE_ID)
- Validate Activation_Code for RemSE_ID
  - *The Activation Code can only be used one time for one activation*
  - *(Option) Validate OTP (if used)*
- <if OK>
  Update Remote-SE User Profile (in Remote-SE System)
  - Store Mobile_ID *(Used for Notification)*
  - *(Option) Store User supplied value for Mobile_PIN*

[6] Register Mobile_ID to Notification Service

[7] Generate Mobile Key (M_Key) using **Fn_Gen_MKey** and store info in Remote-SE User Profile

[8] Return Information to Mobile Payment Application *(SSL/TLS Connection)*
- Mobile Key (M_Key)
- RemSE_ID *(Unique ID to access Remote-SE User Profile)*
- Notification_URL *(Used to connect to Remote-SE System when Notification is received by Mobile Payment Application)*

Set Status of Remote-SE User Profile to **Initialized**

[9] Store Information in encrypted local storage of Mobile Payment Application:
- Mobile Key (M_Key)
- RemSE_ID
- Notification_URL

[10] Wipe Mobile Storage Key (ST_Key)

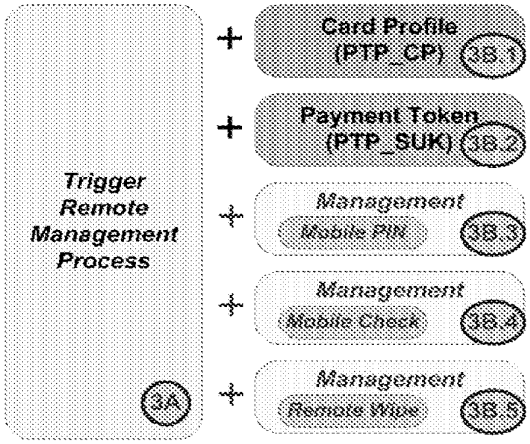
[11] *Event Management (Synchronization between Remote-SE System and Issuer)*
<if OK>
- *Mobile Payment Application ready to be remotely managed*
- *No PayPass transaction allowed at this stage for the selected product (No Payment Credentials delivered)*
- *Definition of rules for remote management*

[12] Mobile Payment Application ready for Remote Management
(UI to inform Consumer)

[13] *Rules to trigger Provisioning process*

©2013 MasterCard – Proprietary and Confidential

Activity #3 – Remote Management



- **Trigger Remote Management**
  - *Activity #3A – Remote Management (Trigger)*
    The Remote Notification process is common to all the Remote Management activities.
- **Remote Management Function**
  - *Activity #3B.1 – Remote Management (PTP_CP)*
    Management of Payment Token Payload (Card Profile)
  - *Activity #3B.2 – Remote Management (PTP_SUK)*
    Management of Payment Token Payload (Single Use key)
  - *Activity #3B.3 – Remote Management (Mobile_PIN)*
    Management of Mobile Payment Application when Mobile_PIN has been changed (e.g. Following the change of Online PIN if this value is used as the value of Mobile_PIN). In this case, all the preloaded Payment Token Payload (Single Use key) must be removed from the Mobile Payment Application.
  - *Activity #3B.4 – Remote Management (Mobile Check)*
    Function to retrieve the current status of the Payment Token Payload (PTP_CP and PTP_SUK) available in the Mobile Payment Application.
    - This function can be used to synchronize the information known by the Remote-SE System (Payment Credentials Management) and the information actually available in the Mobile Payment Application.
    - This also can also be used to extend the Push Model provisioning mechanism and provide the User with the opportunity to send a request to the Remote-SE System (Payment Credentials Management) to get new Single Use Keys.
      In that case the Remote-SE System would use the function to retrieve status information about the Mobile Payment Application. Using that input the Remote-SE could revert to the standard Push mechanisms to deliver new payment credentials to the Mobile Payment Application.
  - *Activity #3B.5 – Remote Management (Remote Wipe)*
    This function wipes all the information stored in the Mobile Payment Application (i.e. Payment Credentials, Mobile Key and any other stored values) while providing the status of the Mobile Payment Application before the removal of credentials is executed by the Mobile Payment Application.

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 134

Activity #3A – Remote Management (Trigger)

App Store | Browser (Mobile or PC) | User | Mobile Device | Mobile Payment Application | Payment Credentials Management (Remote-SE System) | Remote Notification Service (Remote-SE System) | Merchant (PayPass PoS) | Acquirer | Payment Network | Issuer

[1] Trigger to start Remote Management
- From Payment Credentials Management (PCM) using a pre-defined rule
- From Issuer

PCM (Remote-SE System) to identify Remote-SE User Profile to be managed:
- Retrieve RemSE_ID value using **Fn_Get_RemSEID** with info from Issuer (and from PCM)
- Retrieve Remote-SE User Profile using RemSE_ID

[2] Prepare Data for Remote Notification:
- Generate Session_ID *(14 bytes)* using **Fn_Gen_SessID**
  - *(Option) Define Validity (+ Rules in Remote-SE System) for Session_ID*
- Build RemMgt_Info (1 byte) with one of the following functions using **Fn_Gen_RemMgtInfo**
  - ➢ Construct and provision PTP_CP *(Payment Token Payload – Card Profile)*
  - ➢ Construct and provision PTP_SUK *(Payment Token Payload – Single Use Key)*
    - Check if PTPCP_Status of Remote-SE User Profile is **Enabled**, otherwise this function cannot be used before a Card Profile is pushed to the Mobile Payment Application
    - <if OK> Generate PTP_SUK using **Fn_Gen_PTPSUK**
      - Generate Encryption Key (E_Key) using **Fn_Gen_EK**
      - Build Buffer using (Length | PTP_SUK)
      - EPTP_SUK := Encrypt Buffer using E_Key
  - ➢ Mobile Check
  - ➢ *Mobile_PIN Change (e.g. After Online PIN change)*
  - ➢ Remote Wipe Mobile Payment Application Data *(e.g. Reaction after Fraud Detection)*
  - *(This list of remote management actions can be extended)*
- Build Msg using (Length | RemMgt_Info | Session_ID)
  or
  Build Msg using (Length | RemMgt_Info | Session_ID | EPTP_SUK)
- E(Msg) := Encrypt Msg using M_Key
- Build Notification Message using E(Msg)
- Store Session_ID and RemMgt_Info in Remote-SE User Profile
- <if PTP_SUK> Store PTP_SUK and E_Key in Remote-SE User Profile
- Retrieve Mobile_ID from Remote-SE User Profile

[3] Send Notification Message to Mobile_ID

[4] Receive Notification Message with E(Msg)
- (Option) UI (OS Level) to Display Notification Alert

[5] (Optional) Start Mobile Payment Application

[6] (Automatic or Manual [5]) Start Mobile Payment Application
- Integrity Check / Rooted Device Check
  <if NOK> UI to Inform User
- Retrieve Rnd_Storage from Mobile Payment Application Data
- Re-generate Mobile Storage Key (ST_Key) using **Fn_Gen_STKey**
- Retrieve Mobile Key (M_Key) from encrypted local storage
- Decrypt E(Msg) using M_Key
- Validate Length
- <if OK> Validate RemMgt_Info

[7] <if OK> Connect to Remote-SE System using Notification_URL
(SSL/TLS Connection):
- Validate SSL/TLS Connection (Certificates + Common Name)

[8] <if OK> *(SSL/TLS Connection)*:
- Retrieve Session_ID from Msg
  - *(Option) Check Validity*
- <if OK> Generate Authentication_Code using **Fn_Gen_AuthCode**
  - RemSE_ID
  - Session_ID
- Send Authentication Credentials
  - RemSE_ID [Identification]
  - Authentication_Code [Authentication]
- *Wipe Authentication_Code from Memory*

[9] Validate Authentication Credentials:
- Retrieve Session_ID from Remote-SE User Profile (RemSE_ID)
- <if OK> Generate Authentication_Code using **Fn_Gen_AuthCode**
  - RemSE_ID
  - Session_ID
- Check Authentication_Code (Remote-SE System) vs. Authentication_Code (Mobile)
- <if OK> Continue process **[10]** to perform action defined in stored RemMgt_Info

Continue process [10]

©2013 MasterCard – Proprietary and Confidential

Activity #3B.1 – Remote Management (PTP_CP)

App Store | Browser (Mobile or PC) | User | Mobile Device | Mobile Payment Application | Payment Credentials Management (Remote-SE System) | Remote Notification Service (Remote-SE System) | Mastercard PayPass PDS | Acquirer | Payment Network | Issuer

[10] Build **Payment Token Payload - Card Profile (PTP_CP)**
- Get Payment Credentials for Remote-SE User Profile using **Fn_Get_PayCred**
- Generate PTP_CP using **Fn_Gen_PTPCP**
- Build Msg using (Length | PTP_CP)
- Generate Mobile Session Key (MS_Key) using **Fn_Gen_MSKey**
- E(Msg) := Encrypt Msg using MS_Key
- Store PTP_CP in Remote-SE User Profile
- Set PTPCP_Status of Remote-SE User Profile to **Generated**

[11] Return E(Msg) to Mobile Payment Application (SSL/TLS Connection)

[12] Generate Mobile Session Key (MS_Key) using **Fn_Gen_MSKey**

[13] Validate E(Msg).
- Decrypt E(Msg) using MS_Key
- Validate Length
- <If OK> Generate PTP Activation Proof (PTP_ActProof) using **Fn_Gen_PTP_ActProof**
- Store PTC_CP in encrypted local storage
- Remove any available PTC_SUK from encrypted local storage

[14] Build Msg using (Length | PTP_ActProof)
- E(Msg) := Encrypt Msg using MS_Key
- Set PTPCP_Status to **Stored**
- Set PTPSUK_Status to **Empty**
- Set PTPSUK_Counter to **00**

[15] (SSL/TLS Connection)
- Send E(Msg)

[16] Validate E(Msg):
- Decrypt E(Msg) using MS_Key
- Validate Length
- <If OK> Validate PTP Activation Proof (PTP_ActProof) using **Fn_Val_PTP_ActProof**

[17] <If OK> Activate **Payment Token Payload (Card Profile)**
- Set PTPCP_Status of Remote-SE User Profile to **Enabled**
- Set PTPSUK_Status of Remote-SE User Profile to **Empty**
- Set PTPSUK_Counter of Remote-SE User Profile to **00**

[18] *Fraud Management (Systems/Links between Remote-SE System and Issuer)*
<If OK>
- *Card Profile delivered to Mobile Payment Application*
- *No PayPass transaction allowed at this stage for the relevant product (no Single Use Key delivered)*

[19] Inform (Return Code) Mobile Payment Application (SSL/TLS Connection)
Wipe Mobile Session Key (MS_Key)

[20] Analyze Return Code
- <If OK> Set PTPCP_Status to **Enabled** (or **Empty** if failed)

[21] Wipe Mobile Session Key (MS_Key) and Mobile Storage Key (ST_Key)

[22] *Mobile Payment Application ready to receive keys (UI to inform Consumer)*

Activity #3B.2 – Remote Management (PTP_SUK)

App Store | Browser (Portal or PC) | User | Mobile Device | Mobile Payment Application | Payment Credentials Management (Remote-SE System) | Remote Notification Service (Remote-SE System) | MasterCard (PayPass PDS) | Acquirer | Payment Network | Issuer

[10] Retrieve from Remote-SE User Profile the Encryption Key (E_Key) used to protect **Payment Token Payload – Single Use Key (PTP_SUK)**
- Build Msg using (Length | E_Key)
- Generate Mobile Session Key (MS_Key) using **Fn_Gen_MSKey**
- E(Msg) := Encrypt Msg using MS_Key

[11] Return E(Msg) to Mobile Payment Application (SSL/TLS Connection)

[12] Generate Mobile Session Key (MS_Key) using **Fn_Gen_MSKey**

[13] Validate E(Msg):
- Decrypt E(Msg) using MS_Key
- Validate Length
- Retrieve E_Key from Msg
- <If OK> Decrypt EPTP_SUK
  - Validate Length
- **<If OK>** Add PTC_SUK in encrypted local storage with PTPSUK_Status to **Stored**
- Generate PTP Activation Proof (PTP_ActProof) using **Fn_Gen_PTP_ActProof**

[14] Build Msg using (Length | PTP_ActProof)
- E(Msg) := Encrypt Msg using MS_Key

[15] (SSL/TLS Connection)
- Send E(Msg)

[16] Validate E(Msg):
- Decrypt E(Msg) using MS_Key
- Validate Length
- <If OK> Validate PTP Activation Proof (PTP_ActProof) using **Fn_Val_PTP_ActProof**

[17] **<If OK> Activate Payment Token Payload (Single Use Key)**
- Set PTPSUK_Status of Remote-SE User Profile to **Enabled**
- Increment PTPSUK_Counter of Remote-SE User Profile

[18] Fraud Management (Synchronization between Remote-SE System and Issuer)
<If OK>
- Card Profile and Single Use Key(s) delivered to Mobile Payment Application
- PayPass transaction allowed for the selected product

[19] Inform (Return Code) Mobile Payment Application (SSL/TLS Connection)
Wipe Mobile Session Key (MS_Key)

[20] Analyze Return Code
- **<If OK>** Set PTPSUK_Status to **Enabled** of previously added PTP_SUK (or remove if failed)
- Increment PTPSUK_Counter

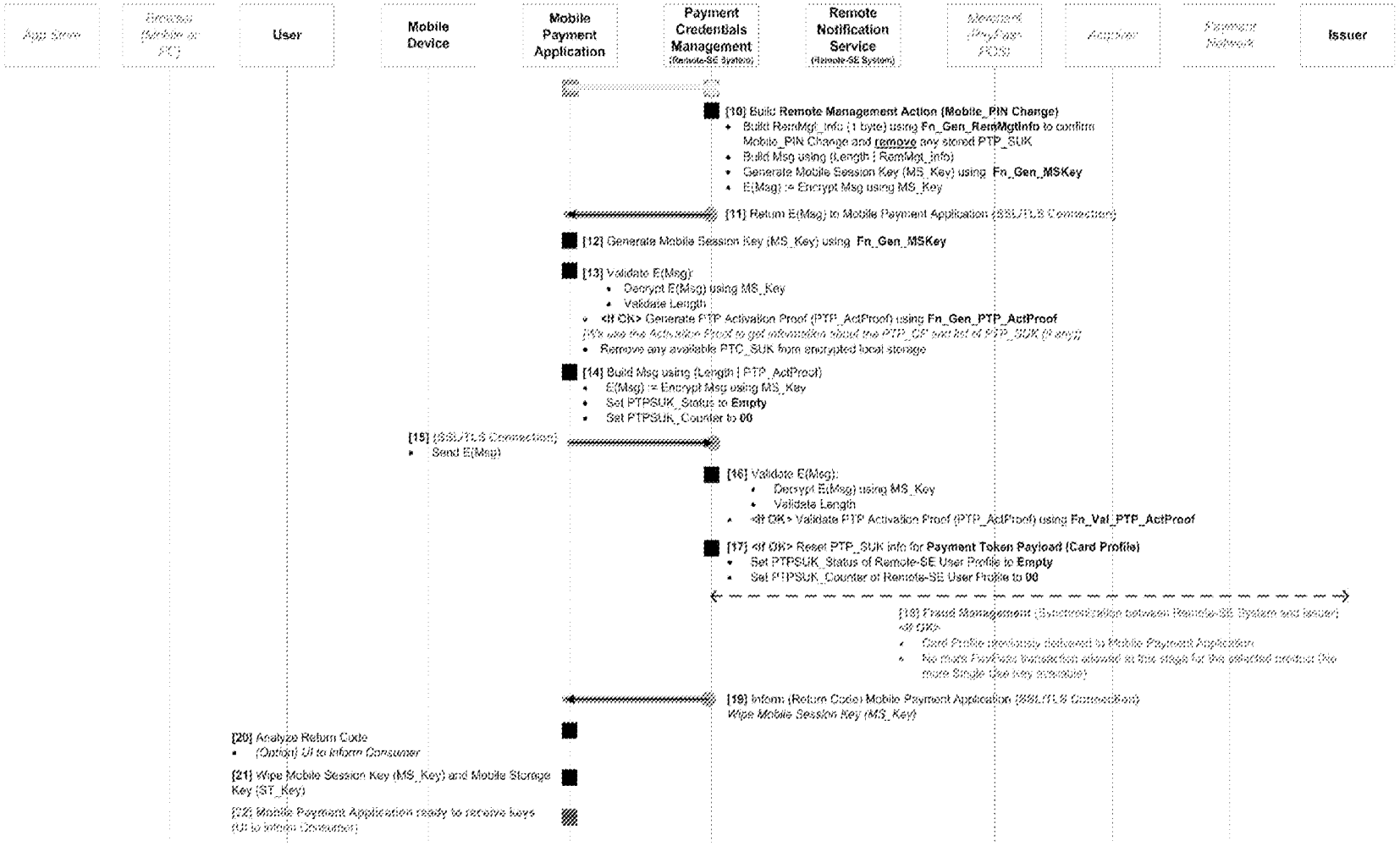
[21] Wipe Mobile Session Key (MS_Key) and Mobile Storage Key (ST_Key)

[22] Mobile Payment Application ready for Contactless Payment
(UI to Inform Consumer)

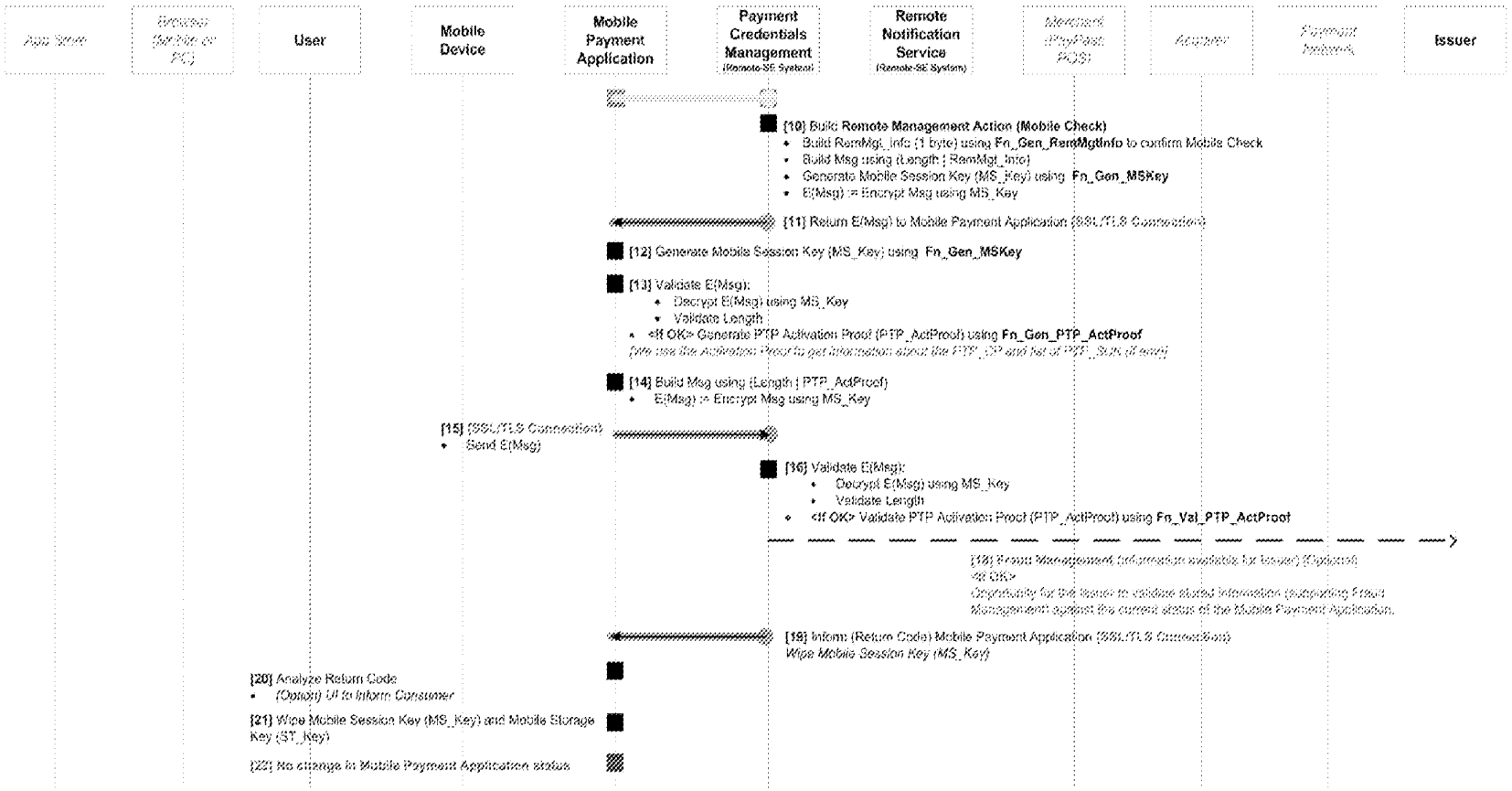
©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 137

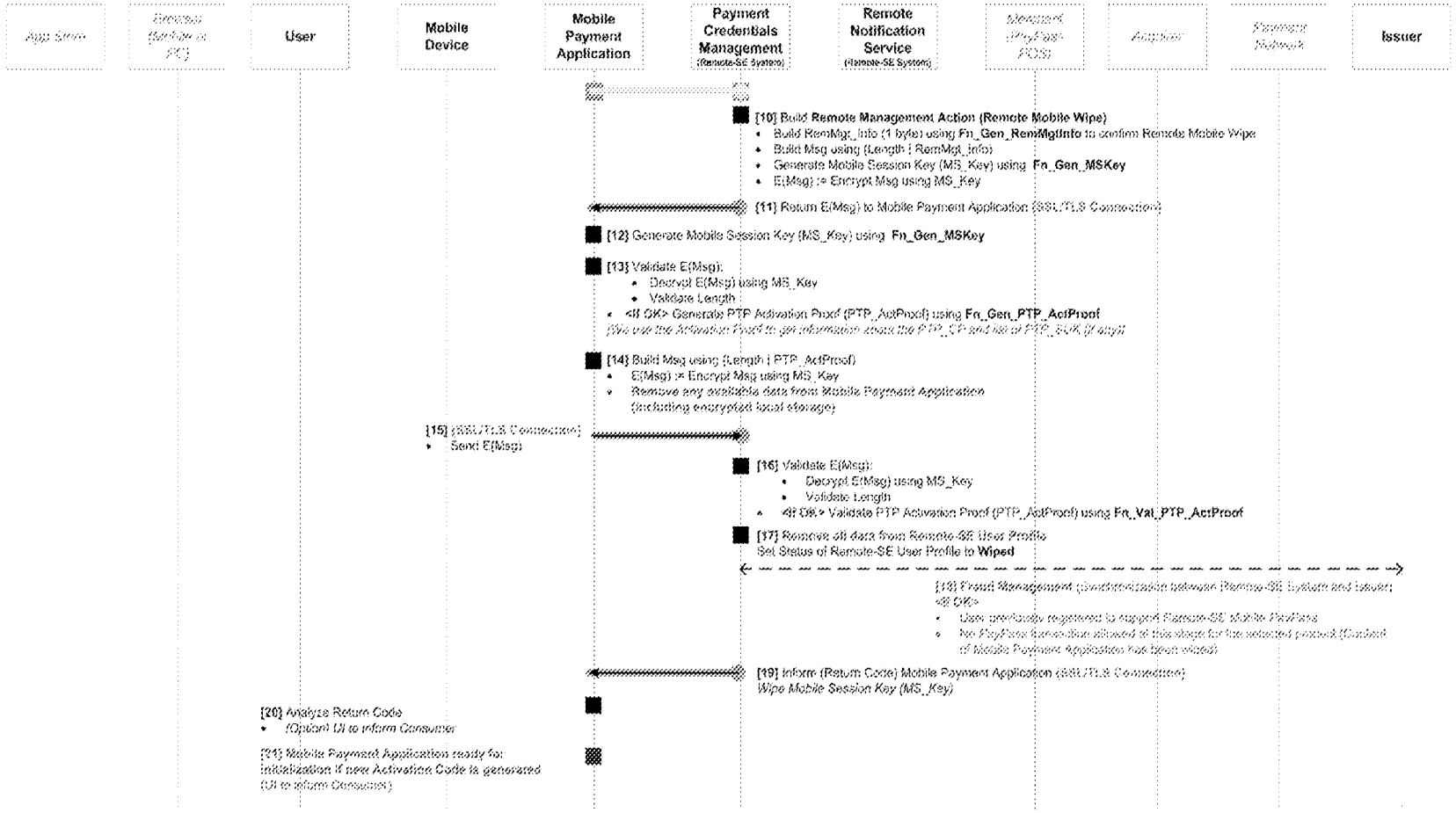
Activity #3B.3 – Remote Management (Mobile_PIN)



Activity #3B.4 – Remote Management (Mobile Check)



Activity #3B.5 – Remote Management (Remote Wipe)



Activity #4 – Payment (*PayPass* transaction)

App Store | Browser (Mobile or PC) | User | Mobile Device | Mobile Payment Application | Payment Credentials Management (Remote-SE System) | Remote Administration Service (Remote-SE System) | Merchant (*PayPass* POS) | Acquirer | Payment Network | Issuer

[1] Start Mobile Payment Application

[2] Mobile Payment Application
- Integrity Check / Rooted Device Check
- <If NOK> UI to Inform User
- Retrieve Rnd_Storage from Mobile Payment Application Data
- Re-generate Mobile Storage Key (ST_Key) using **Fn_Gen_STKey**
- Retrieve Payment Credentials (PTP_CP and PTP_SUK) from encrypted local storage *(Use PTP_SUK with lowest ATC Value)*
- Check availability and status
- UI to Inform User
- <If OK> Continue

[3] Execute **Remote-SE Payment Transaction Flow**
(Option) Use a « Simplified » Mobile *PayPass* (MPP) software implementation using a limited support of MPP Core Specification (i.e. without all the features not applicable in the context of Remote-SE Mobile *PayPass*)

[4] Depending on the outcome of the transaction, update PTPSUK_Status and PTPSUK_Counter

[5] Standard process for *PayPass* Transaction (Merchant ⇔ Acquirer)

[6] Standard process for *PayPass* Transaction (Acquirer ⇔ Payment Network)

[7] Standard process for *PayPass* Transaction (Payment Network ⇔ Issuer)

[8] Issuer ready to validate *PayPass* transaction

[9] CVC3 Validation using **Fn_Val_CVC3**
Or AC Validation using **Fn_Val_AC**

[10] Complete Standard process for *PayPass* Transaction (Issuer ⇔ Payment Network ⇔ Acquirer ⇔ Merchant)

[11] Fraud Management (Synchronization between Remote-SE System and Issuer)
- *Card Profile (previously delivered to Mobile Payment Application)*
  - *If Single Use Key(s) available, PayPass transaction allowed for the selected product*
  - *Otherwise, no PayPass transaction allowed at this stage for the selected product (No Single Use Key available)*

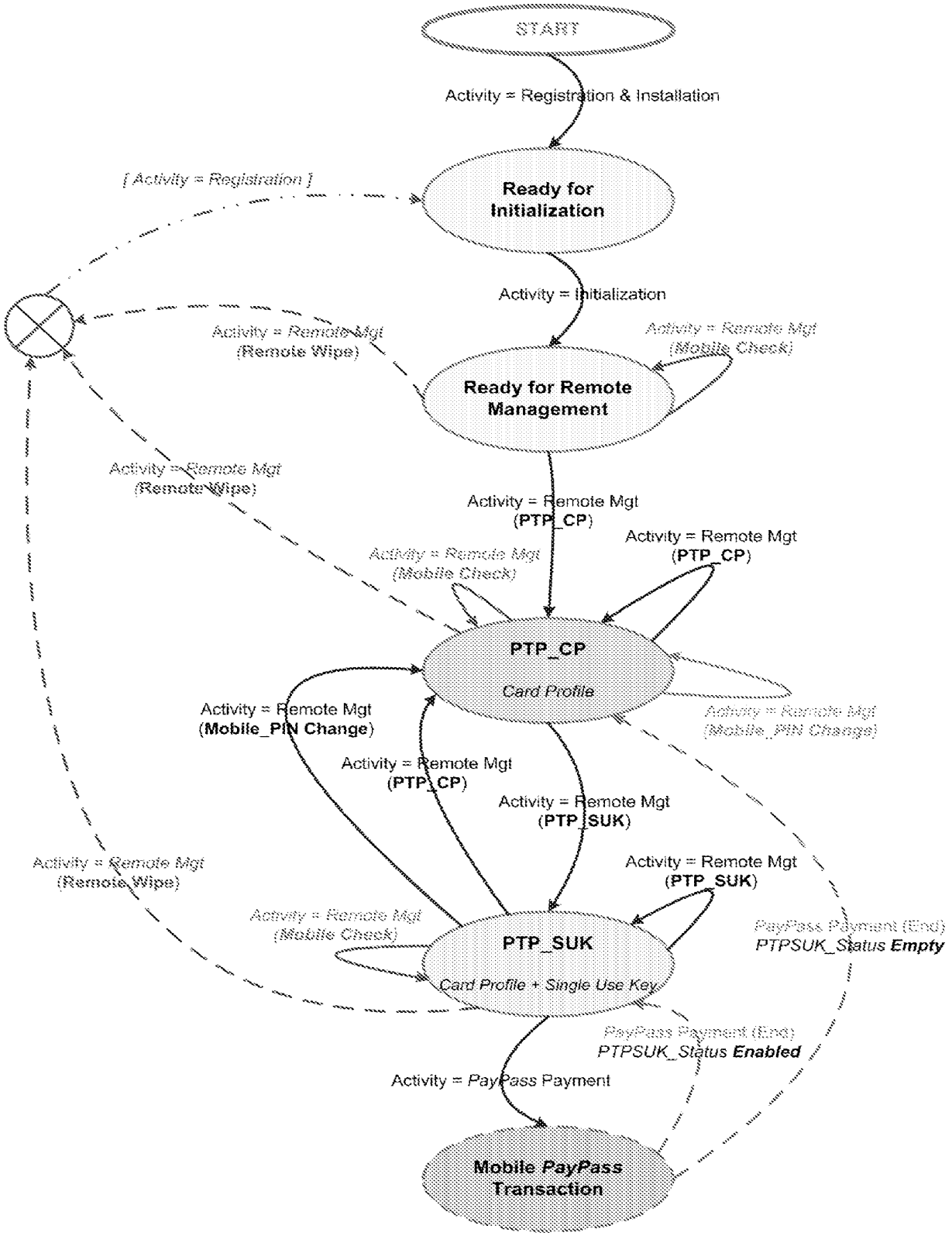
[12] Depending on the outcome of the transaction, update PTPSUK_Status and PTPSUK_Counter

[13] Wipe Mobile Storage Key (ST_Key)

[14] Mobile Payment Application ready for Contactless Payment
&
Mobile Payment Application ready to receive keys
(UI to Inform Consumer)

Remote-SE Mobile *PayPass* State Machine (Activities)

This figure describes the state machine used to support the Remote-SE Mobile *PayPass* solution.



IPR2025-01152
Apple EX1008 Page 142

Key Management

This section explains the management of the following keys used to support Remote-SE Mobile *PayPass*:

- *Mobile Key* (*M_Key*)
- *Session Key* (*MS_Key*)
- *Storage Key* (*ST_Key*)
- *Payment Key* (*SUK_Key*)
- *Encryption Key* (*E_Key*)

Additional keys and certificates are used:

- SSL/TLS connection used at time of registration.
  Those keys and certificates are under the control of the Issuer (and the Remote-SE system).
- Connection between the Remote-SE System and the Remote Notification Service.
  The cryptography used to support that connection is dependent of the Mobile Vendor.
- Connection between the Remote Notification Service and the Mobile Payment Application.
  The cryptography used to support that connection is dependent of the Mobile Vendor.
- SSL/TLS connection between the Mobile Payment Application and the Remote-SE System (Payment Credentials Management):
  - When using the Remote-SE System to initialize the Mobile Payment Application (*Initialization_URL*)
  - When connecting to the Remote-SE System to process a remote notification message (*Notification_URL*).
- Payment credentials (PKI information available in PTP_CP – *M/Chip*) used to support *Local Data Authentication*.

Mobile Key

- The Mobile Key (*M_Key*) is generated by the Remote-SE System.
- The key is delivered to the Mobile Payment Application during the *Activity #2 – Initialization*.
- The key is shared between the Remote-SE System and the Mobile Payment Application.
- The key is used to encrypt the message sent to the Mobile Payment Application using the *Remote Notification*.
- The key is also used to derive a Mobile Session Key (*MS_Key*) with the session identifier (*Session_ID*) delivered as part of the encrypted message using the Remote Notification Service.

Session Key

- The Mobile Session Key (*MS_Key*) is derived from the Mobile Key (*M_Key*) using the session identifier (*Session_ID*) as diversifier.
- The session identifier is delivered as part of the encrypted message sent to the Mobile Payment Application using the Remote Notification Service.

Storage Key

- The Storage Key (ST_Key) is generated using a mechanism described in *Encrypted Local Database in Mobile Payment Application Data*.
- The Mobile Payment Application uses three factors to generate the key:
  - Information from the Mobile Device
  - Information from the Mobile Payment Application Code
  - A random value (*Rnd_Storage*) generated by the Mobile Payment Application
- The Storage Key is used by the Mobile Payment Application to support a local encrypted database.

IPR2025-01152
Apple EX1008 Page 143

Payment Key

- Remote-SE Mobile *PayPass* solution uses the concept of Single Use Key (*SUK_Key*) valid for one single Mobile *PayPass* Transaction.
- A Payment Token Payload (*PTP_SUK*) contains one key which can be used either to generate an Application Cryptogram (AC) when using *PayPass* M/Chip or to generate a dynamic Card Validation Code (CVC3) when using *PayPass* Mag Stripe.
    - Remote-SE System uses a standard mechanism for Card Master Key (CMK) generation. The algorithm uses the Issuer Master Key (IMK) with the PAN and PSN as diversifiers.
    - Remote-SE System uses a standard EMV CSK Session Key derivation for generation of the payment Session Key.
      The algorithm uses the Card Master Key (CMK) with the Application Transaction Counter as a diversifier.
    - When using *PayPass* M/Chip, this process is standard.
    - When using *PayPass* Mag Stripe, an additional session key derivation process must be added to the key management (and CVC3 validation process) as the Remote-SE Mobile *PayPass* solution denies any provisioning of a Card Master Key (CMK) to the Mobile Payment Application.
      The Mobile Payment Application must only be provisioned with payment keys which can only be used one single time.
- In addition to that process, the Single Use Key is combined with the *Mobile PIN* value using a ***Error! Reference source not found.*** described in *Appendix B – Payment Transaction Flow*.
    - If the Mobile PIN value was correct, then the cryptogram generation process used a valid key and the cryptogram validation should be successful.
    - Otherwise a wrong key was used and the cryptogram validation process should fail.

Encryption Key

- The Single Use Key (*SUK_Key*) is delivered encrypted (*EPTP_SUK*) to the Mobile Payment Application using a remote notification message (Refer to *Activity #3A – Remote Management (Trigger)* and *Activity #3B.2 – Remote Management (PTP_SUK)*).
- A random encryption key (*E_Key*) is generated by the Remote-SE System to encrypt the *PTP_SUK* containing the *SUK_Key* value.
- This key will be delivered to the Mobile Payment Application (using a SSL/TLS channel and encrypted using the Mobile Session Key (*MS_Key*)) after a successful *Mutual Authentication* between the Remote-SE System (server) and the Mobile Payment Application (client).
- The Mobile Payment Application uses this key to retrieve the value of the *PTP_SUK*.

Functions

This section lists the functions used to support Remote-SE Mobile *PayPass*:

- Key Management
  - *Fn_Gen_STKey*
  - *Fn_Gen_MKey*
  - *Fn_Gen_MSKey*
  - *Fn_Gen_EK*
- Identifiers Management
  - *Fn_Gen_RemSEID*
  - *Fn_Get_RemSEID*
  - *Fn_Gen_SessID*
- Remote Management
  - *Fn_Gen_RemMgtInfo*
- Client Authentication
  - *Fn_Gen_AuthCode*
- Payment Token Payload Management
  - *Fn_Get_PayCred*
  - *Fn_Gen_PTPCP*
  - *Fn_Gen_PTPSUK*
- Activation (and status information)
  - *Fn_Gen_PTP_ActProof*
  - *Fn_Val_PTP_ActProof*

Note **The information listed as reserved for future use (RFU) is reserved for MasterCard use Only.**

Fn_Gen_STKey

This function is used to generate a Storage Key (*ST_Key*) using:

1. Information about the Mobile Device (IMEI Value[10] or Serial Number[11]).
   A hash function (SHA256) is computed over that information to generate a 32 bytes data element.
2. Information (16 bytes) embedded in the code of the Mobile Payment Application.
3. A random value (16 bytes) generated by the Mobile Payment Application (*Rnd_Storage*).

- The three components are concatenated to build a message.
- A hash function (SHA256) is computed over that message to build a 32 bytes key (*ST_Key*).

The key is used (AES256) by the Mobile Payment Application to support a local encrypted database.

Fn_Gen_MKey

The Remote-SE System (Payment Credentials Management) generates the Mobile Key (*M_Key*) (32 bytes) using either a random key generation process or a key derivation process using some parameters from the Remote-SE User Profile.

Fn_Gen_MSKey

This function generates the Mobile Session Key (*MS_Key*) (32 bytes) using the following key derivation process:

- The *Session_ID* (14 bytes) is used to build a message.
- A HMAC_SHA256 function is computed using Mobile Key (*M_Key*) over that message to build a 32 bytes key (*MS_Key*).

The Mobile Session Key (*MS_Key*) (32 bytes) is used (AES256) to protect the exchange of information between the Mobile Payment Application and the Remote-SE System.

[10] Available in Mobile Phones but not in Tablets or Mobile Devices without telephony support (e.g. Wi-Fi only).
[11] Usually available in Tablets

IPR2025-01152
Apple EX1008 Page 145

Fn_Gen_EK

This function generates a Random Encryption Key (*E_Key*) (32 bytes) used to protected the delivery of the *PTP_SUK*.
The PTP_SUK is sent encrypted (*EPTP_SUK*) over the *Remote Notification* channel.

The encryption key (*E_Key*) (32 bytes) is delivered encrypted to the Mobile Payment Application using the *SSL/TLS Communication* layer after successful *Mutual Authentication*.
The encryption (AES256) is performed using the Mobile Session Key (*MS_Key*).

Fn_Gen_RemSEID

This function generates a unique identifier (*RemSE_ID*) (32 bytes) linked with the identifier (*User_ID*) provided by the Issuer to the User.

A hash (SHA256) is computed over a message built using the concatenation of the following set of data:

- *User_ID* (32 bytes)
- Random information (16 bytes) (to be stored in the Remote-SE User Profile[12])

The hash value is the unique identifier (*RemSE_ID*) (32 bytes).

Fn_Get_RemSEID

This function is used to retrieve the value *RemSE_ID* when providing the *User_ID*.

Fn_Gen_SessID

This function generates the Session Identifier (*Session_ID*) (14 bytes) using the following template:

- B1 Version Control & Format (1 byte) as defined in Table 3
- B2-B11 Random Value (10 bytes)
- B12-B14 Expiry Date (3 bytes using YYMMDD format)

Table 3—Session ID (Byte 1 – Version Control & Format)

b8	b7	b6	b5	b4	b3	b2	b1	Description
x	x	x						Version ◆ '001' – Reserved for MasterCard ◆ '010' – Reserved for MasterCard ◆ '011' – Product Description (v3.0 and v3.1) ◆ Other values are RFU
			x					◆ b5=1: Control on the expiry of the Remote Notification message ◆ b5=0: No control of expiry. Expiry Date value is set to random (3 bytes)
				x	x	x	x	RFU

Fn_Gen_RemMgtInfo

This function generates the Remote Management Information (*RemMgt_Info*) (1 byte) using the format defined in Table 4.

Table 4—Remote Management Information

b8	b7	b6	b5	b4	b3	b2	b1	Description
x	x	x						Version ◆ '001' – Reserved for MasterCard ◆ '010' – Reserved for MasterCard ◆ '011' – Product Description (v3.0 and v3.1) ◆ Other values are RFU
			x	x	x	x	x	Remote Management Function ◆ 00001 (PTP_CP) ◆ 00010 (PTP_SUK) ◆ 11100 (Mobile Check) ◆ 11101 (Mobile_PIN Changed) ◆ 11111 (Remote Wipe) ◆ Other values are RFU

[12] It is an Issuer decision to replicate this information in the User Profile managed by the Issuer.

Fn_Gen_AuthCode

This function generates an Authentication Code (24 bytes).

A hash (SHA256) is computed over a message built using the concatenation of the following set of data:

- *RemSE_ID* (32 bytes)
- *Session_ID* (14 bytes)

The hash value is truncated (24 leftmost bytes) to deliver the authentication code (*Authentication_Code*).

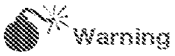
Fn_Get_PayCred

This function collects all the Payment Credentials to be included in the Payment Token Payload (Card Profile) (*PTP_CP*):

- Information common to *PayPass* Mag Stripe and *PayPass* M/Chip
- Information specific to *PayPass* Mag Stripe
- Information specific to *PayPass* M/Chip, including the certificate required to support the *Local Data Authentication*

Fn_Gen_PTPCP

This function creates the "perso profile" required to deliver all the *PTP_CP* to the Mobile Payment Application.
The detailed content of the "perso profile" is out of scope of this document.



It is strongly recommended to consider the use of a « Simplified » Mobile *PayPass* (MPP) software implementation using a limited support of MPP Core Specification [MPP_SU101].
The limited support means that we don't need to support the features not applicable in the context of Remote-SE Mobile *PayPass*.

The specification of the « Simplified » Mobile *PayPass* (MPP) software implementation is out of scope of this document.

Fn_Gen_PTPSUK

This function generates the value of *PTP_SUK* using:

- Truncated Hash (SHA 256) (24 leftmost bytes) computed over the content of (*PTP_CP*)
- ATC (2 bytes)
- A payment Single Use Key (*SUK_Key*) (16 bytes)

Fn_Gen_PTP_ActProof

This function generates the Payment Token Payload Activation Proof (*PTP_ActProof*) used to activate PTP_CP or PTP_SUK.

- When provisioning a new PTP_CP, it contains:
  - Truncated Hash (SHA 256) computed over the old PTP_CP (if available)
  - Truncated Hash (SHA 256) computed over PTP_CP
  - The number of PTP_SUK available (*PTPSUK_Counter*)
  - The list of ATCs of those PTP_SUK
- When provisioning a new PTP_SUK, it contains:
  - Hash (SHA 256) computed over PTP_CP
  - The number of PTP_SUK available (*PTPSUK_Counter*)
  - The list of ATCs of those PTP_SUK (if any)

The function concatenates the following data elements to deliver the Activation Proof (*PTP_ActProof*):

- Version Control & Format (1 byte) as defined in Table 5
- Truncated hash (24 leftmost bytes) (Old PTP_CP) (if available)
- Truncated hash (24 leftmost bytes) (PTP_CP)
- Counter (PTP_SUK) (1 byte)
- List of concatenated ATC of PTP_SUK (2 bytes | 2 bytes | … | 2 bytes) (if # PTP_SUK <> 00) (n * 2 bytes)

Table 5—Activation Proof (Version Control & Format)

b8	b7	b6	b5	b4	b3	b2	b1	Description
x	x	x						Version ♦ '001' – Reserved for MasterCard ♦ '010' – Reserved for MasterCard ♦ '011' – Product Description (v3.0 and v3.1) ♦ Other values are RFU
			x					♦ b5=1: PTP_CP (Old) included ♦ b5=0: PTP_CP (Old) not included
								♦ b4=1: PTP_SUK information available ♦ b4=0: PTP_SUK information not available
				x	x	x	x	RFU

Fn_Val_PTP_ActProof

This function is used to validate the value of *PTP_ActProof* provided by the Mobile Payment Application.

IPR2025-01152
Apple EX1008 Page 148

Payment Functions

This section lists the functions used to support payment related processes of Remote-SE Mobile *PayPass*:

- Cryptogram Validation
  - *Fn_Val_CVC3*
  - *Fn_Val_AC*
- Local Data Authentication
  - *Fn_Gen_LDA*
- Single Use Key Generation
  - *Fn_Gen_SUK*

Fn_Val_CVC3

This function is used by the Issuer to validate a CVC3 generated by Remote-SE Mobile *PayPass*.
The CVC3 validation process must integrate an additional session key derivation step instead of using directly the CVC3 key (KD_{CVC3}).
EMV CSK algorithm (using the ATC as a diversifier) is used to generate the session key.

Fn_Val_AC

This function is used by the Issuer to validate an Application Cryptogram generated by Remote-SE Mobile *PayPass*.
A standard AC validation process (using EMV CSK as session key derivation algorithm) is used when using Remote-SE Mobile *PayPass*.

Fn_Gen_LDA

This function is used to generate the data elements required to support *Local Data Authentication*.

Fn_Gen_SUK

This function is used to generate a Single Use Key (Payment key) that will be used to generate a cryptogram (AC or CVC3).
Remote-SE Mobile *PayPass* uses the EMV CSK algorithm for session key derivation for both *PayPass* M/Chip and *PayPass* Mag Stripe (using the Issuer updated cryptographic process).

The Remote-SE System must support an additional step to protect the key to be provisioned to the Mobile Payment Application.

- The session payment key must be combined with the *Mobile PIN* using the ***Error! Reference source not found.***.
- The result of this computation is the *SUK_Key*.

Data Elements

This section lists the data elements used to support Remote-SE Mobile *PayPass*:

- Keys
  - *ST_Key*
  - *M_Key*
  - *MS_Key*
  - *E_Key*
  - *SUK_Key*
- Identifiers
  - *User_ID*
  - *RemSE_ID*
- Mobile PIN
  - *Mobile PIN*
- Remote Notification
  - *Mobile_ID*
- Local Encrypted Database
  - *Rnd_Storage*
- Session Identifier
  - *Session_ID*
- Activation and Authentication
  - *Activation_Code*
  - *Authentication_Code*
  - *PTP_ActProof*
- URL
  - *Initialization_URL*
  - *Notification_URL*
- Remote Management
  - *RemMgt_Info*
- Card Profile
  - *PTP_CP* (*Common*, *Mag Stripe* and *M/Chip*)
  - *PTPCP_Status*
- Single Use Key
  - *PTP_SUK*
  - *PTPSUK_Status*
  - *PTPSUK_Counter*
  - *EPTP_SUK*

ST_Key

The Mobile Storage Key (32 bytes – AES256) generated by the Mobile Payment Application using *Fn_Gen_STKey*.

M_Key

The Mobile Key (32 bytes – AES256) generated by the Remote-SE System using *Fn_Gen_MKey*.

MS_Key

The Mobile Session Key (32 bytes – AES256) derived by both the Remote-SE System and the Mobile Payment Application using *Fn_Gen_MSKey*.

E_Key

A random Encryption Key (32 bytes – AES256) generated by the Remote-SE System using *Fn_Gen_EK*.

SUK_Key

The Single Use Payment Key (16 bytes – 3DES) generated by both the Remote-SE System and the Issuer using *Fn_Gen_SUK*.

User_ID

A unique identifier provided by the Issuer to the User.
The format of the User Identifier is defined by the Issuer.

In order to use a fixed length (32 bytes) data element, a hash (SHA256) is computed over a message built using the "human" value of the User Identifier as shared between the Issuer and the User.

RemSE_ID

A unique identifier (32 bytes) defined by the Remote-SE System (Payment Credentials Management) using *Fn_Get_RemSEID* to uniquely identify a Remote-SE User Profile.

Mobile PIN

The concept of Mobile PIN is defined in *Appendix B – Payment Transaction Flow* (*Mobile PIN*).
It is a design decision (also driven by requirements for the User Interaction) to have a Mobile PIN up to 8 digits

Mobile_ID

A technical identifier generated by the Mobile Payment Application when registering to the Remote Notification Service using standard mechanisms (APIs) delivered by Mobile Vendors (e.g. Google Cloud Messaging for Android (GCM), Apple Push Notification Service (APNS)…).

Rnd_Storage

A random value (16 bytes) generated by Mobile Payment Application as part of the process used to build the encryption key (*MS_Key*) used to protect the local storage of information.

Session_ID

The session identifier generated by the Remote-SE System (Payment Credentials Management) using *Fn_Gen_SessID*.

Activation_Code

The value provided by the Remote-SE System (Payment Credentials Management) to activate the Remote-SE User Profile during the initialization of the Remote-SE Mobile *PayPass* Mobile Payment Application.
This value is linked with the *User_ID* and the *RemSE_ID*.

Authentication_Code

The Authentication Code generated by the Mobile Payment Application (or the Remote-SE System) using *Fn_Gen_AuthCode*.

PTP_ActProof

The proof for activation of *PTP_CP* or *PTP_SUK* generated by the Mobile Payment Application using *Fn_Gen_PTP_ActProof*.

Initialization_URL

The URL (and Common Name) used by the Mobile Payment Application to connect to the Remote-SE System (Payment Credentials Management) for initialization.

Notification_URL

The URL (and Common Name) used by the Mobile Payment Application to connect to Remote-SE System (Payment Credentials Management) for upon reception of a remote notification message.

RemMgt_Info

The description of the operation to be executed as part of the Remote Management of the Mobile Payment Application using the format defined in Table 4 used by *Fn_Gen_RemMgtInfo*.

PTP_CP

The Payment Token Payload (Card Profile) contains:

- Information *Common* to *PayPass* Mag Stripe and *PayPass* M/Chip
- Information specific to *PayPass Mag Stripe*
- Information specific to *PayPass M/Chip*, including the certificate required to support the *Local Data Authentication*

IPR2025-01152
Apple EX1008 Page 151

Common

The list of Common data elements should include, but is not restricted to, the following data elements:

- PAN ('5A' – Var up to 10 bytes), PSN ('5F34' – 1 byte)
- AID (5-16 bytes)
- Application Control ('D7' – 4 bytes)
- AFL ('94' – Var up to 252 bytes)
- AIP ('82' – 2 bytes)
- Track 1 Data ('56' – Var up to 76 bytes), Track 2 Data ('9F6B' – Var up to 19 bytes)
- FCI Template (PPSE) ('6F' – Var)
- FCI Template (*PayPass*) ('6F' – Var)
- Card Layout Description (DF47 – Var) [Part 1, Part 2 and Part 3]

Mag Stripe

The list of Mag Stripe data elements should include, but is not restricted to, the following data elements:

- NATC Track 1 ('9F64' – 1 byte), NATC Track 2 ('9F67' – 1 byte)
- PCVC3 Track 1 ('9F62' – 6 bytes), PCVC3 Track 2 ('9F65' – 2 bytes)
- PUNATC Track 1 ('9F63' – 6 bytes), PUNATC Track 2 ('9F66' – 2 bytes)
- IVCVC3 Track 1[13] ('DC' – 2 bytes), IVCVC3 Track 2[14] ('DD' – 2 bytes)
- CIAC – Decline On PPMS ('DF4F' – 2 bytes)
- UDOL ('9F69' – 15 bytes)

M/Chip

The list of M/Chip data elements should include, but is not restricted to, the following data elements:

- CIAC – Decline On ARQC ('DF5D' – 3 bytes), CIAC – Decline On Offline Only ('DF5E' – 3 bytes), CIAC – Decline On Online Capable ('DF45' – 3 bytes), CIAC – Go Online ('DF46' – 3 bytes)
- CDOL1 ('8C' – Var up to 252 bytes), CDOL2 ('8D' – Var up to 252 bytes)
- Application Currency Code ('9F42' – 2 bytes), Application Currency Exponent ('9F44' – 1 byte)
- Application Effective Date ('5F25' – 3 bytes), Application Expiration Date ('5F24' – 3 bytes)
- CVN (1 byte)
- CVM List ('8E' – 10-252 bytes)
- IAC – Default ('9F0D' – 5 bytes), IAC – Denial ('9F0E' – 5 bytes), IAC – Online ('9F0F' – 5 bytes)
- PDOL ('9F38' – Var)
- Certification Authority Public Key Index ('8F' – 1 byte)
- ICC Public Key Certificate ('9F46' – N_I bytes), ICC Public Key Exponent ('9F47' – 1-3 bytes), ICC Public Key Remainder ('9F48' – $N_{IC} - N_I + 42$ bytes), ICC Private Key
- Issuer Public Key Certificate ('90' – N_{CA} bytes), Issuer Public Key Exponent ('9F32' – 1-3 bytes), Issuer Public Key Remainder ('92' – $N_I - N_{CA} + 36$ bytes)
- Signed Dynamic Application Data ('9F4B' – N_{IC} bytes), Static Data Authentication Tag List (9F4A' – Var)

PTPCP_Status

The status of *PTP_CP* is coded using 1 byte.
Two distinct values are used, respectively one in the Mobile Payment Application and one in the Remote-SE System (Payment Credentials Management):

- Empty (00000000)
- Generated (00000001)
- Stored (00000010)
- Enabled (00000011)
- Other values are RFU

[13] As Mobile PIN is always used and not controlled by the Mobile Payment Application, both *IVCVC3 Track 1* and PIN IVCVC3 Track 1 ('DF43') have the same value when considering Remote-SE Mobile *PayPass*.
[14] The same remark applies for Track 2 (PIN IVCVC3 Track 1 – 'DF44').

IPR2025-01152
Apple EX1008 Page 152

PTP_SUK

The Payment Token Payload (Single Use Key) linked with a *PTP_CP.*
The format of the data element is defined in *Fn_Gen_PTPSUK*:

- Truncated Hash (SHA 256) (24 leftmost bytes) computed over the content of (*PTP_CP*)
- ATC (2 bytes)
- A payment Single Use Key (*SUK_Key*) (16 bytes)

PTPSUK_Status

The status of *PTP_SUK* coded using 1 byte.
Two distinct values are used, respectively one in the Mobile Payment Application and one in the Remote-SE System (Payment Credentials Management):

- Empty (00000000)
- Generated (00000001)
- Stored (00000010)
- Enabled (00000011)
- Other values are RFU

PTPSUK_Counter

The number of *PTP_SUK*. The value is coded using 1 byte.

EPTP_SUK

The encrypted value of *PTP_SUK.*

IPR2025-01152
Apple EX1008 Page 153

Appendix B – Payment Transaction Flow

Overview

This appendix presents the following items:

- *PayPass Transaction*
  This section presents the concepts for support of a Mobile *PayPass* transaction using Remote-SE Mobile *PayPass*.
  It provides a functional description of the expected transaction flow.
- *Mobile PIN*
  This section presents the concepts of Mobile PIN.
  It defines the XOR method used to protect the access to the Single Use Key and provides a list of test vectors.
  It explains how the PIN Verification Value (PVV) must be handled when using Remote-SE Mobile *PayPass*.
- *Local Data Authentication*
  This section reminds the need for Mobile CVM and introduces the concept of Local Data Authentication (LDA).
  It explains the Use of Virtual PAN.

IPR2025-01152
Apple EX1008 Page 154

PayPass Transaction

This section presents the *Concepts* for support of a Mobile *PayPass* transaction using Remote-SE Mobile *PayPass*.
It provides a functional description of the expected *Transaction Flow*.

Concepts

- The Mobile Payment Application must support both Mobile *PayPass* Mag Stripe and Mobile *PayPass* M/Chip transaction flows.
- The *PayPass* transaction flow uses the payment credentials provisioned by the Remote-SE System to the Mobile Payment Application.
- The payment credentials are available in the Card Profile (*PTP_CP*) and in the Single Use Key (*PTP_SUK*)
  - The Payment Token Payload (Card Profile) contains:
    - Information *Common* to *PayPass* Mag Stripe and *PayPass* M/Chip
    - Information specific to *PayPass Mag Stripe*
    - Information specific to *PayPass M/Chip*, including the certificate required to support the *Local Data Authentication*
  - The Single Use Key contains:
    - The Application Transaction Counter (ATC)
    - The payment key to be used to generate the cryptogram (AC or CVC3)
- The Mobile CVM is supported when using v3 Reader.

Transaction Flow

The diagram overleaf presents the functional description of the expected transaction flow.

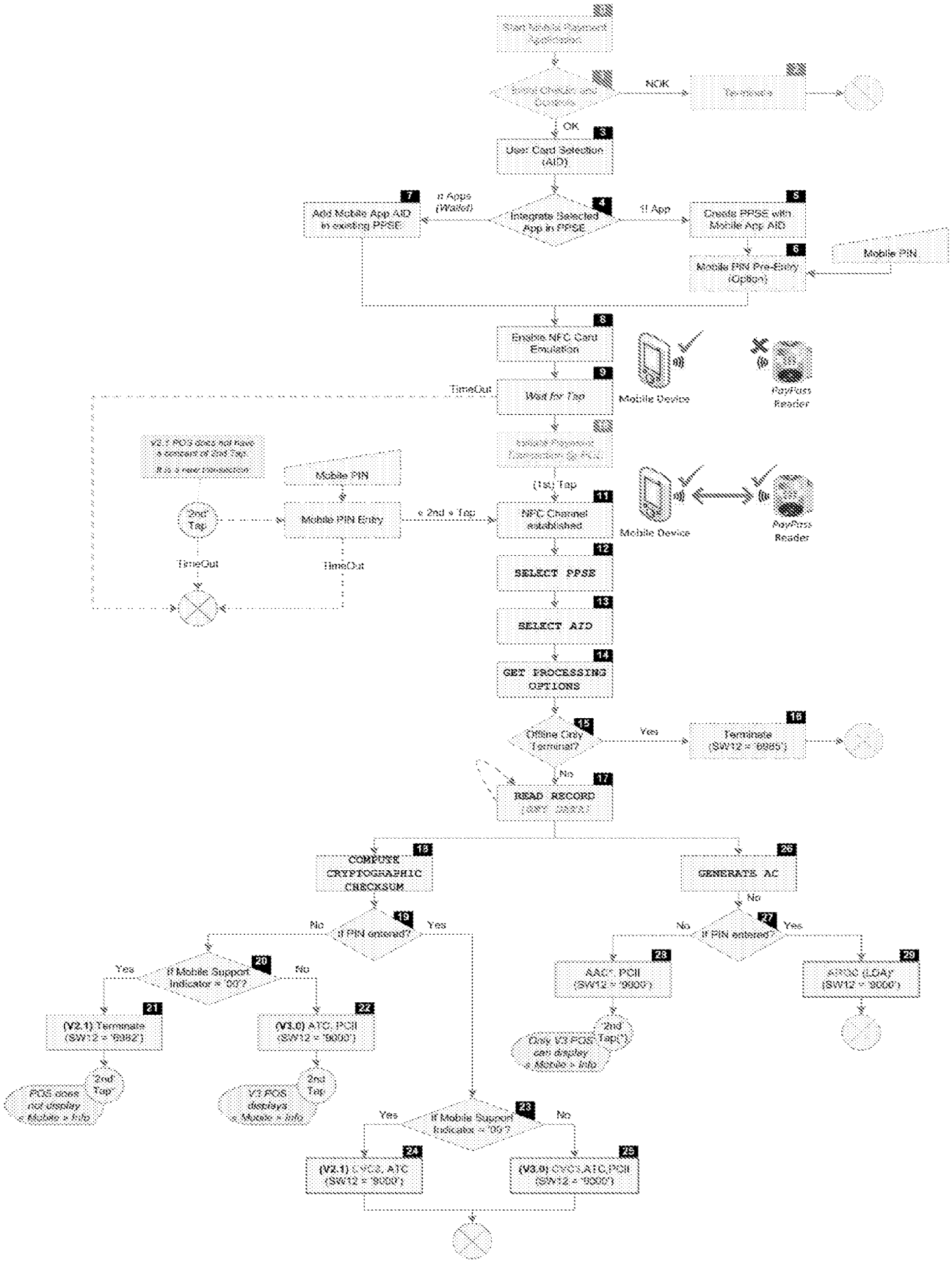
- The Mobile Payment Application can be:
  - A single "card".
    The PPSE has to be set with the AID of the "card".
  - Embedded in a Wallet.
    The AID must be added to the list of AID managed by the PPSE.
- The transaction flows supports Pre-PIN entry.
  If the Mobile PIN is not provided before the first Tap, it will have to be provided before a second Tap as Remote-SE Mobile *PayPass* is PIN Always.

Warning

It is strongly recommended to consider the use of a « Simplified » Mobile *PayPass* (MPP) software implementation using a limited support of MPP Core Specification [MPP_SU101].
The limited support means that we don't need to support the features not applicable in the context of Remote-SE Mobile *PayPass*.

The specification of the « Simplified » Mobile *PayPass* (MPP) software implementation is out of scope of this document.

IPR2025-01152
Apple EX1008 Page 155


NOK
OK
User Card Selection (AID)
Add Mobile App AID in existing PPSE
n Apps (Wallet)
Integrate Selected App in PPSE
!! App
Create PPSE with Mobile App AID
Mobile PIN
Mobile PIN Pre-Entry (Option)
Enable NFC Card Emulation
TimeOut
Wait for Tap
Mobile Device
PayPass Reader
Mobile PIN
(1st) Tap
2nd Tap
Mobile PIN Entry
« 2nd » Tap
NFC Channel established
TimeOut
TimeOut
SELECT PPSE
SELECT AID
GET PROCESSING OPTIONS
Offline Only Terminal?
Yes
No
READ RECORD
COMPUTE CRYPTOGRAPHIC CHECKSUM
GENERATE AC
If PIN entered?
No
Yes
If Mobile Support Indicator = '00'?
Yes
No
(V2.1) Terminate (SW12 = '6982')
(V3.0) ATC, PCII (SW12 = '9000')
2nd Tap
POS does not display « Mobile » Info
V3 POS displays « Mobile » Info
AAC*, PCII (SW12 = '9000')
APDC (LDA)* (SW12 = '9000')
Only V3 POS can display « Mobile » Info
(V2.1) CVC3, ATC (SW12 = '9000')
(V3.0) CVC3,ATC,PCII (SW12 = '9000')

Mobile PIN

This section presents the *Concepts* of Mobile PIN.
It defines the ***Error! Reference source not found.*** used to protect the access to the Single Use Key and provides a list of ***Error! Reference source not found.***.
It explains how the *PIN Verification Value* (PVV) must be handled when using Remote-SE Mobile *PayPass*.

Concepts

When using Remote-SE Mobile *PayPass*, the payment key (16 bytes) delivered to the Mobile Payment Application is combined with the Mobile PIN value using a ***Error! Reference source not found.***.

The basic principles are:

- The User must always provide a Mobile PIN value at time of a *PayPass* transaction when using Remote-SE.
    - It is an Issuer decision to synchronize the Mobile PIN value with the Online PIN value. This solution will provide the best user experience when double PIN entry is required (High value transactions when using v2.1 POS).
    - In that case the Security Rules and Procedures (SR&P) rule (4.2) for Online PIN applies: "*PINs must be numeric, alphabetic, or alphanumeric, and up to six digits long*". It does mean that we can consider that the Mobile PIN can be up to six digits long.
    - The storage on the Mobile PIN in the Remote-SE System is addressed by the clarification note of PCI DSS requirement 3.2 where it is permissible for Issuers and companies that support issuing services to store sensitive authentication data if there is a business justification and the data is stored securely.
    - Otherwise, it is a design decision (also driven by requirements for the User Interaction) to have a Mobile PIN up to 8 digits.
- There is no offline validation of the Mobile PIN value.
- The Mobile PIN value is used by the Mobile Payment Application to rebuild a key using data ("protected" Single Use Key) stored in the local encrypted database.
    - The Mobile Payment Application will always generate a cryptogram even if the Mobile PIN value provided by the User is wrong
    - There is no direct means for the Mobile Payment Application to detect <u>offline</u> that the Mobile PIN is wrong unless the payment transaction is sent <u>online</u> to the Issuer for authorization.
    - Using that mechanism, the Issuer has a direct control on the number of PIN tries performed by the User.
    - If the Mobile PIN value was correct, then the cryptogram generation process used a valid key and the cryptogram validation should be successful.
      Otherwise a wrong key was used and the cryptogram validation process should fail.
- The Mobile Payment Application must make sure that the Mobile PIN is deleted as soon as it is no longer required by the application to minimize the window in which a memory dump might be used to retrieve sensitive credentials.

©2013 MasterCard – Proprietary and Confidential

IPR2025-01152
Apple EX1008 Page 157

PIN Verification Value

The PIN Verification Value (PVV) is the result of a cryptographic function and the value can be disclosed without compromising the PIN value.
The PVV can be stored in Issuer Discretionary Data of a Mag Stripe or on Issuer Host.
The Card Profile (PTP_CP) contains track data which is considered as static data (i.e. not PIN related)
The Payment Token (PTP_SUK) contains a Single Use Key protected using the ***Error! Reference source not found.***

When using the Remote-SE Mobile *PayPass* solution described in this document, any change of the value of the Mobile PIN mandates all the loaded PTP_SUK to be deleted and new PTP_SUK to be provisioned to the Mobile Payment Application. There is no need to change the PTP_CP.

When using PVV with a synchronization of PIN value between Online PIN and Mobile PIN, it would mean that the PVV stored in the Discretionary Data of Mag Stripe would need to be updated when Online PIN is changed.
As a consequence the value of Track Data needs to be changed and therefore a replacement of the Card Profile (PTP_CP) would be required when Online PIN (= Mobile PIN) is changed.

It is a design decision to not support PVV in Issuer Discretionary Data of a Mag Stripe for Online PIN validation when using Remote-SE Mobile *PayPass*.
The PVV (or reference PIN value) must be retrieved from the Issuer Host or the component responsible for Online PIN verification.

Local Data Authentication

This section reminds the *Need for Mobile CVM* and introduces the *LDA Concept*.
It explains the *Use of Virtual PAN*.

Need for Mobile CVM

The expected acceptance of the Remote-SE Mobile *PayPass* is the following for a *PayPass* Reader:

	Hard limit country[15]		**Soft limit** country[16]	
	< v3.0 Reader	v3.x Reader	< v3.0 Reader	v3.x Reader
Transaction Amount **below** CVM limit	*(Technical solution equivalent to Issuer PIN Always)* 1			
Transaction Amount **above** CVM limit	Transaction does not start	Mobile CVM 1	(Mobile CVM) 2 + Online PIN or Signature	Mobile CVM 1

- When using V3 Reader, Mobile CVM prevents "Double PIN entry" (in Mobile and at POS) **[1]**.
  - When using *PayPass* M/Chip, Mobile CVM requires the support of Offline CAM.
  - When using *PayPass* Mag Stripe, Mobile CVM is the standard configuration with Remote-SE Mobile *PayPass*.
- When using v2.1 Reader, Mobile CVM is not supported by the Reader.
  - For low value transaction, there is one single PIN entry on Mobile **[1]**
  - For high value transaction in hard limit countries, the transaction does not start
  - For high value transaction in soft limit countries, the "Double PIN Entry" cannot be avoided.
    There is a PIN entry on Mobile, a Tap of the Mobile Device and then an additional CVM at the POS **[2]**.The additional CVM can be Online PIN (PIN Entry on PED) or Signature.

The support of Offline CAM, as defined today with CDA, would introduce a potential critical security issue:

[15] [PPReq] (*Hard Limit)* A maximum transaction amount is set and cardholder verification (CVM) is never required below this limit.
[16] [PPReq] (*Soft Limit)* Transactions over a given value require cardholder verification (CVM).

IPR2025-01152
Apple EX1008 Page 158

- One of the characteristics of Remote-SE Mobile *PayPass* is the potential disclosure of payment credentials when using a software-based solution.
- A valid private key (even a key valid for one single transaction) with a valid certificate that would be compromised (because it is not stored in a SE) would allow the creation of rogue cards (using genuine credentials) that pass offline CAM validation.

Such a solution must NOT be supported from a security point of view.

Without the support of CDA, the support of Mobile CVM is not possible.
As a consequence, the Remote-SE Mobile *PayPass* would be handled by the Reader (POS) as a Card and not as a Mobile Device. It would mean that "Double PIN entry" would be the rule even when using a v3.x Reader.
The concept of Local Data Authentication (LDA) is used to deliver Offline CAM support (~ a CDA like solution) when using Remote-SE Mobile *PayPass*.

LDA Concept

The concept of Local Data Authentication (LDA) requires:

- One RSA Key Pair and Certificate as part of the Card Profile (*PTP_CP – M/Chip*)
- The certificate must use a specific model:
  - LDA swaps the meaning of Effective Date and Expiry date
  - The Expiry Date of Application set in the "past" (e.g. 10/11/2012)
  - The Effective Date[17] of set in the future (e.g. Provisioning data of *PTP_CP* + 3 years)
  - The ICC PKCert Expiry Date is set to the same Effective Date
  - The mandatory list of tags defined in the SDA Tag List (Key, PAN, Expiry, Exponent…) must be available and used.
  - Set IACs (Application Expired / Application Not yet Effective) for Online / Default. It will force Online and decline Offline.
  - At POS level, we use the following:
    - Application Expired bit set in TVR
    - Application Not yet effective bit set in TVR

Warning

The solution relies on the following strong assumption: a POS will not decline transaction when "Expired" but will send the transaction for Online authorization.

The Offline Only terminals should decline the transaction and it is the expected behavior when using Remote-SE Mobile *PayPass* solution.

The LDA solution should be evaluated as part of a technology trial of the Remote-SE Mobile *PayPass* solution.

Use of Virtual PAN

Virtual PAN can be used as a mechanism to provide segregation between the PAN (and PSN) value delivered to the Mobile Payment Application as part of a Card Profile (PTP_CP) from the value actually used at level of the Issuer System (or any on-behalf services) for the authorization process.

The use of Virtual PAN can allow an Issuer to mitigate the risk of any misuse of a PAN value for CNP eCommerce transactions.
On the other hand, the use of Virtual PAN and the frequency of replacement of the PAN value must be carefully considered in the context of Remote-SE Mobile *PayPass*.

The support of Mobile CVM (when using V3 Readers) requires use of *Local Data Authentication*.

- As part of the LDA support, a RSA key pair and a certificate must be generated.
- The Static Data to be Authenticated is data signed by the Issuer Private Key in the Signed Static Application Data.
- This static data is also used to produce the Public Key Certificate.
- The PAN and PSN values are among the data that must[18] be signed.

It does mean that any change of the PAN (and PSN) value mandates a new certificate to be issued (and a new RSA key pair to be generated).

[17] / !\ Rules for February 29.
[18] M/Chip Requirements, Oct 2012, Chapter 4 – Data Requirements: Static Data to be Authenticated

IPR2025-01152
Apple EX1008 Page 159

That information is part of the Card Profile (PTP_CP) associated with a set of Payment Tokens (Single Use Keys).
As a direct consequence, any PAN/PSN replacement in the context of Virtual PAN triggers:

- Key management to be performed by the Remote-SE System (SDA generation, certificate issuance and RSA Key Pair generation).
- The provisioning of a new Card Profile (PTP_CP) to the Mobile Payment Application
- The removal of any existing Payment Token (Single Use Key) stored in the local encrypted database of the Mobile Payment Application
- The provisioning of at least one new Payment Token (PTP_SUK) to support a *PayPass* transaction

The frequency used to change the Virtual PAN has a direct impact on the viability of the Remote-SE Mobile *PayPass* solution if an Issuer would decide to change PAN after a limited number of transactions (e.g. less than 5-10 transactions, the worst case would be after every transaction).

Warning

- Remote-SE Mobile *PayPass* can support the use of Virtual PAN.
- The frequency for PAN (or PSN) replacement must be carefully defined to deliver a viable solution.

••• End of document •••

IPR2025-01152
Apple EX1008 Page 160

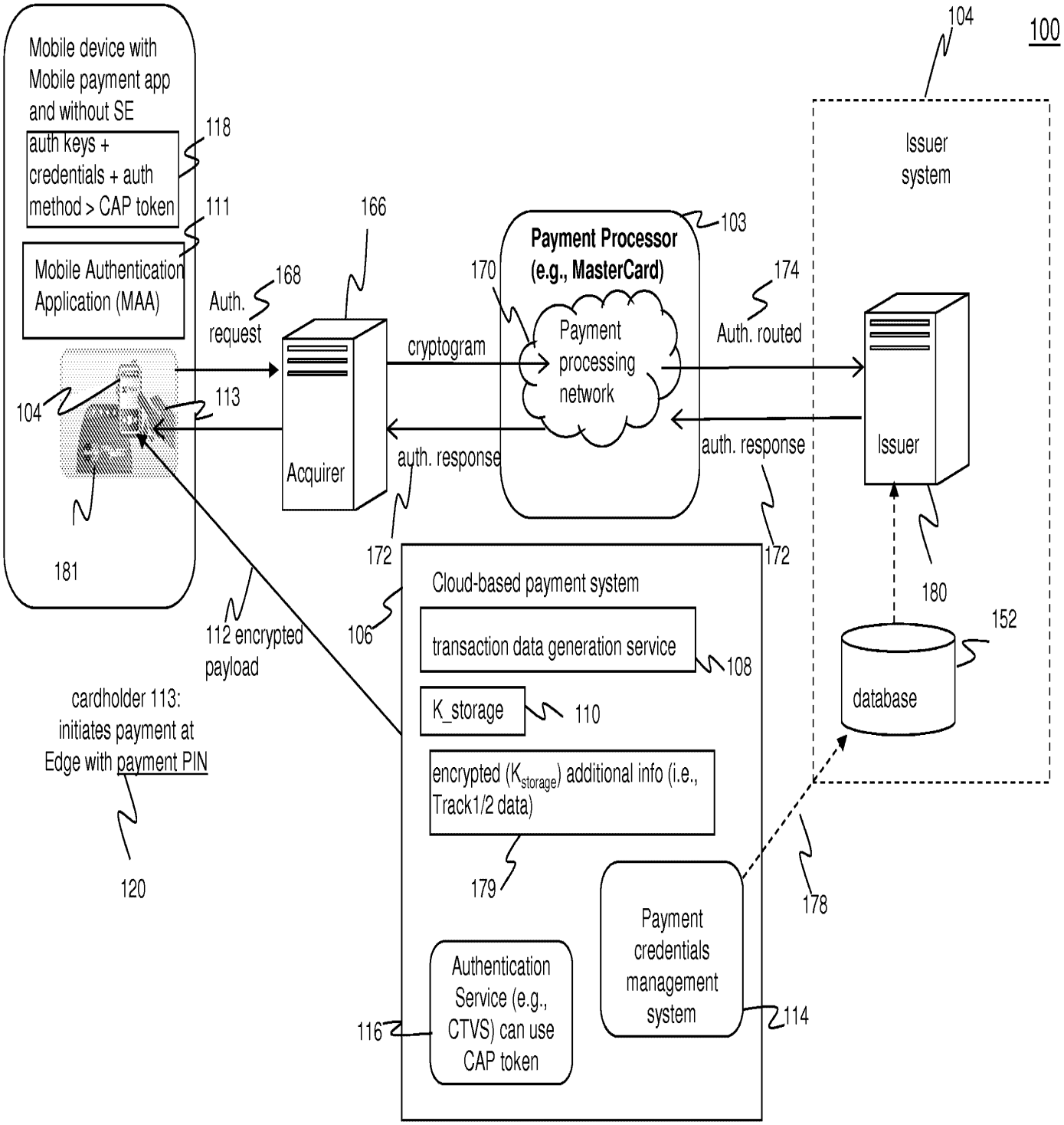

100
104
Mobile device with Mobile payment app and without SE
auth keys + credentials + auth method > CAP token
118
111
Mobile Authentication Application (MAA)
Auth. request
168
166
Acquirer
113
104
181
cryptogram
Payment Processor (e.g., MasterCard)
103
170
Payment processing network
174
Auth. routed
Issuer system
Issuer
auth. response
auth. response
172
172
180
152
database
Cloud-based payment system
106
transaction data generation service
108
K_storage
110
encrypted ($K_{storage}$) additional info (i.e., Track1/2 data)
179
Payment credentials management system
114
178
Authentication Service (e.g., CTVS) can use CAP token
116
112 encrypted payload
cardholder 113: initiates payment at Edge with payment PIN
120

FIG. 1

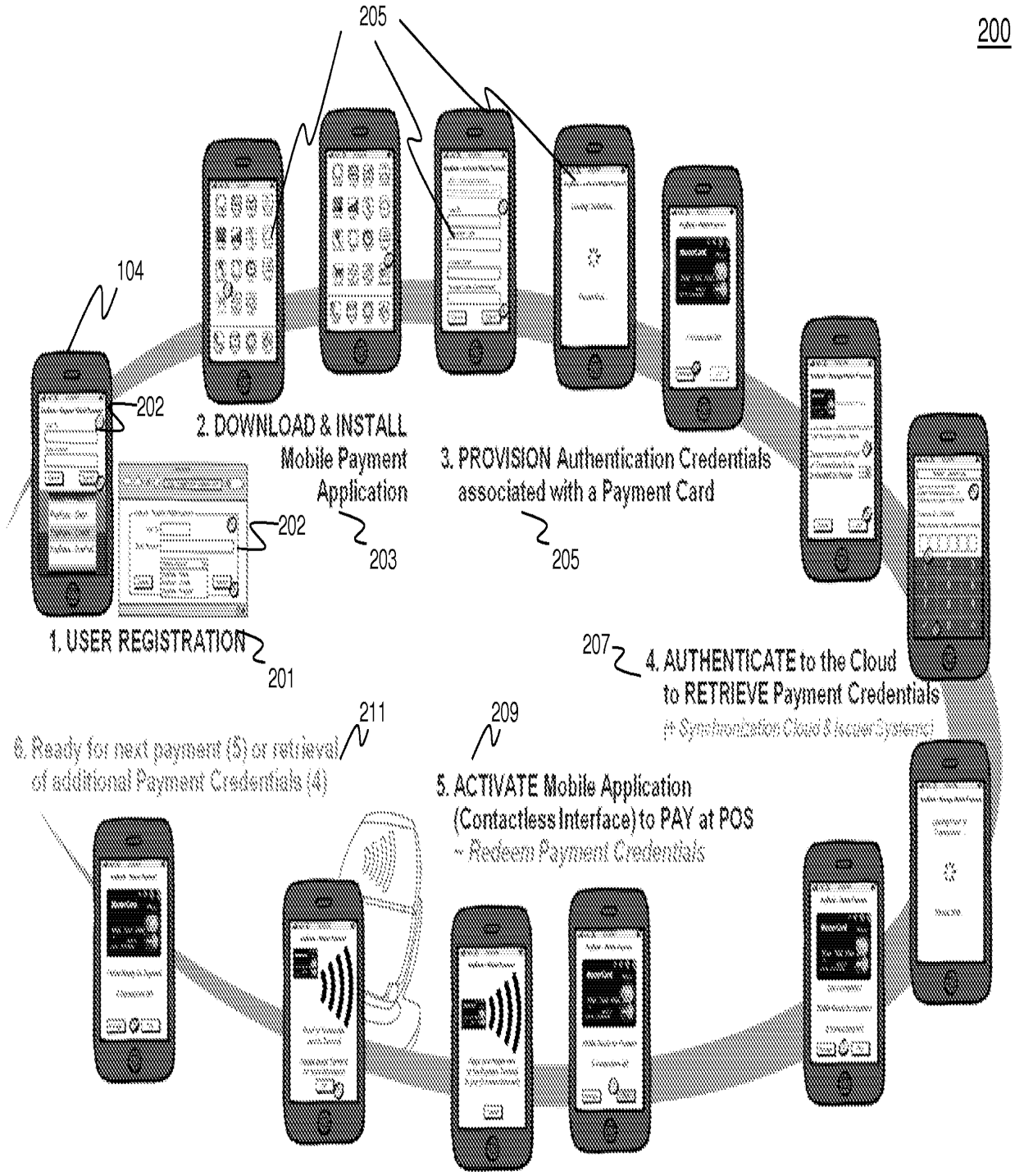

200
205
104
202
2. DOWNLOAD & INSTALL
Mobile Payment
Application
203
3. PROVISION Authentication Credentials
associated with a Payment Card
205
202
1. USER REGISTRATION
201
207
4. AUTHENTICATE to the Cloud
to RETRIEVE Payment Credentials
211
209
6. Ready for next payment (5) or retrieval
of additional Payment Credentials (4)
5. ACTIVATE Mobile Application
(Contactless Interface) to PAY at POS
~ Redeem Payment Credentials

FIG. 2A

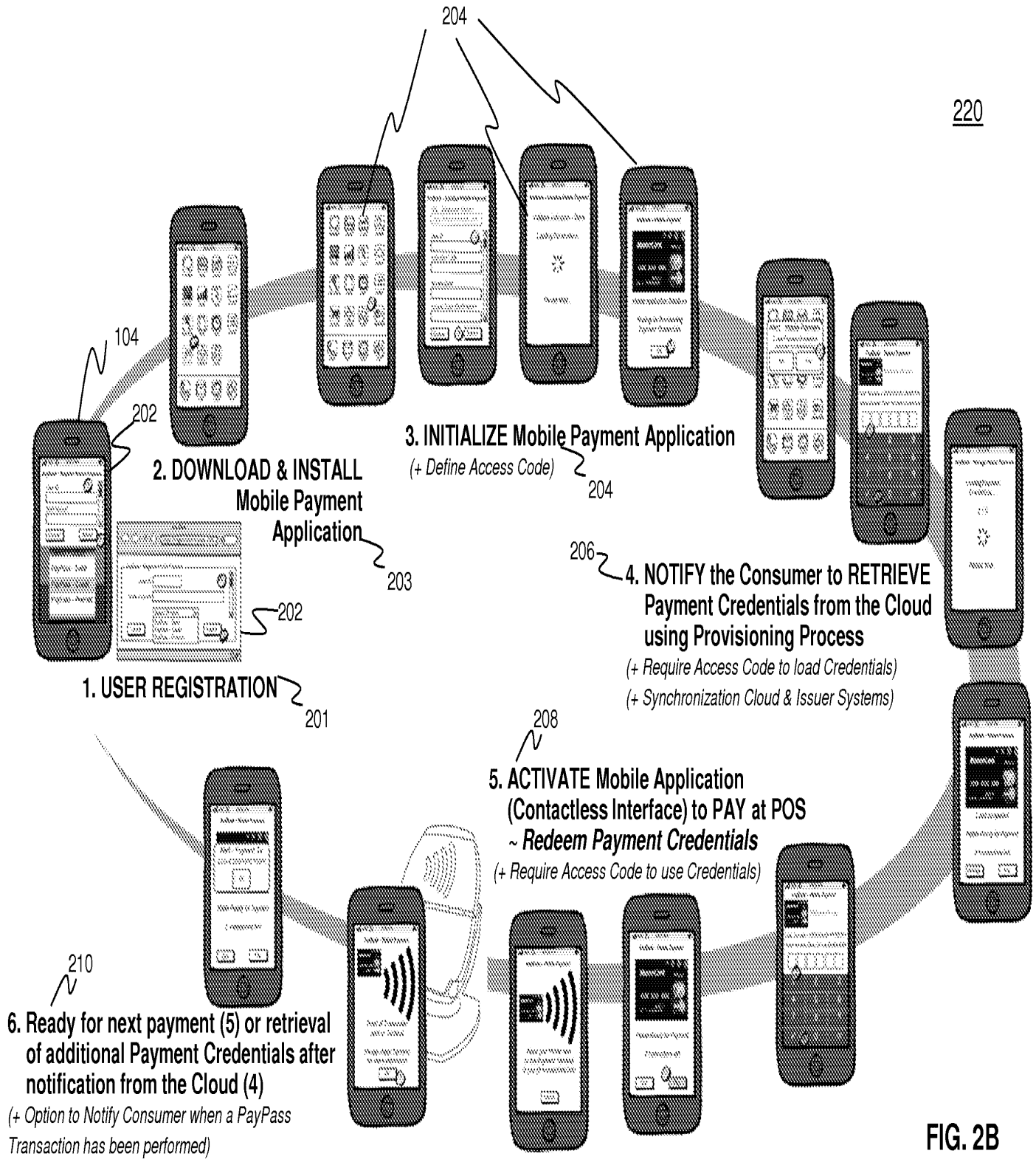

204
220
104
202
2. DOWNLOAD & INSTALL
Mobile Payment
Application
203
3. INITIALIZE Mobile Payment Application
(+ Define Access Code)
204
206
4. NOTIFY the Consumer to RETRIEVE
Payment Credentials from the Cloud
using Provisioning Process
(+ Require Access Code to load Credentials)
(+ Synchronization Cloud & Issuer Systems)
202
1. USER REGISTRATION
201
208
5. ACTIVATE Mobile Application
(Contactless Interface) to PAY at POS
~ Redeem Payment Credentials
(+ Require Access Code to use Credentials)
210
6. Ready for next payment (5) or retrieval
of additional Payment Credentials after
notification from the Cloud (4)
(+ Option to Notify Consumer when a PayPass
Transaction has been performed)

FIG. 2B

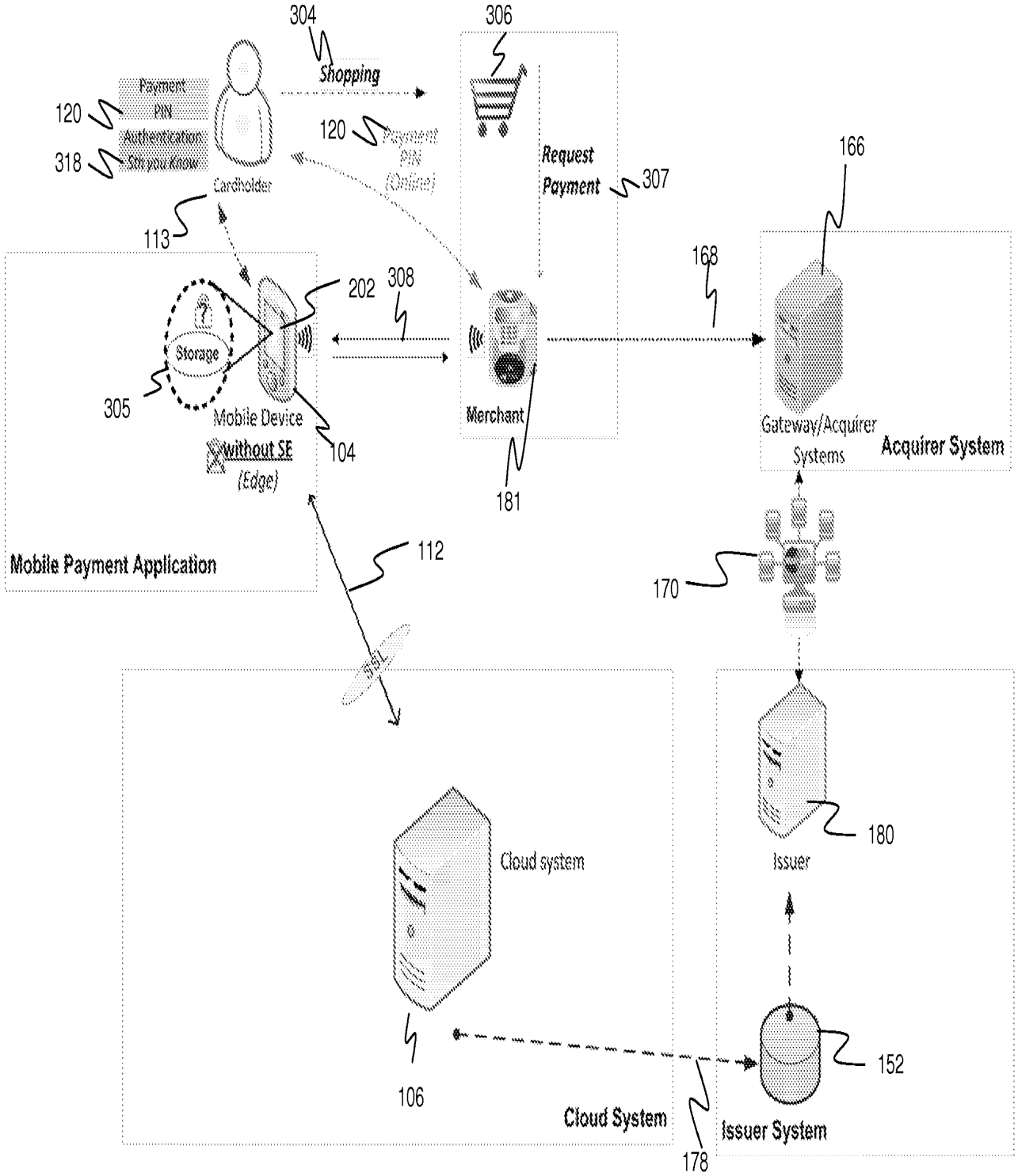

304
306
Shopping
Payment
PIN
120
Authentication
318
Cardholder
120
Payment
PIN
(Online)
Request
Payment
307
166
113
168
202
308
Storage
305
Mobile Device
without SE
(Edge)
104
Merchant
181
Gateway/Acquirer
Systems
Acquirer System
Mobile Payment Application
112
170
NFC
Cloud system
180
Issuer
106
152
Cloud System
Issuer System
178

FIG. 3

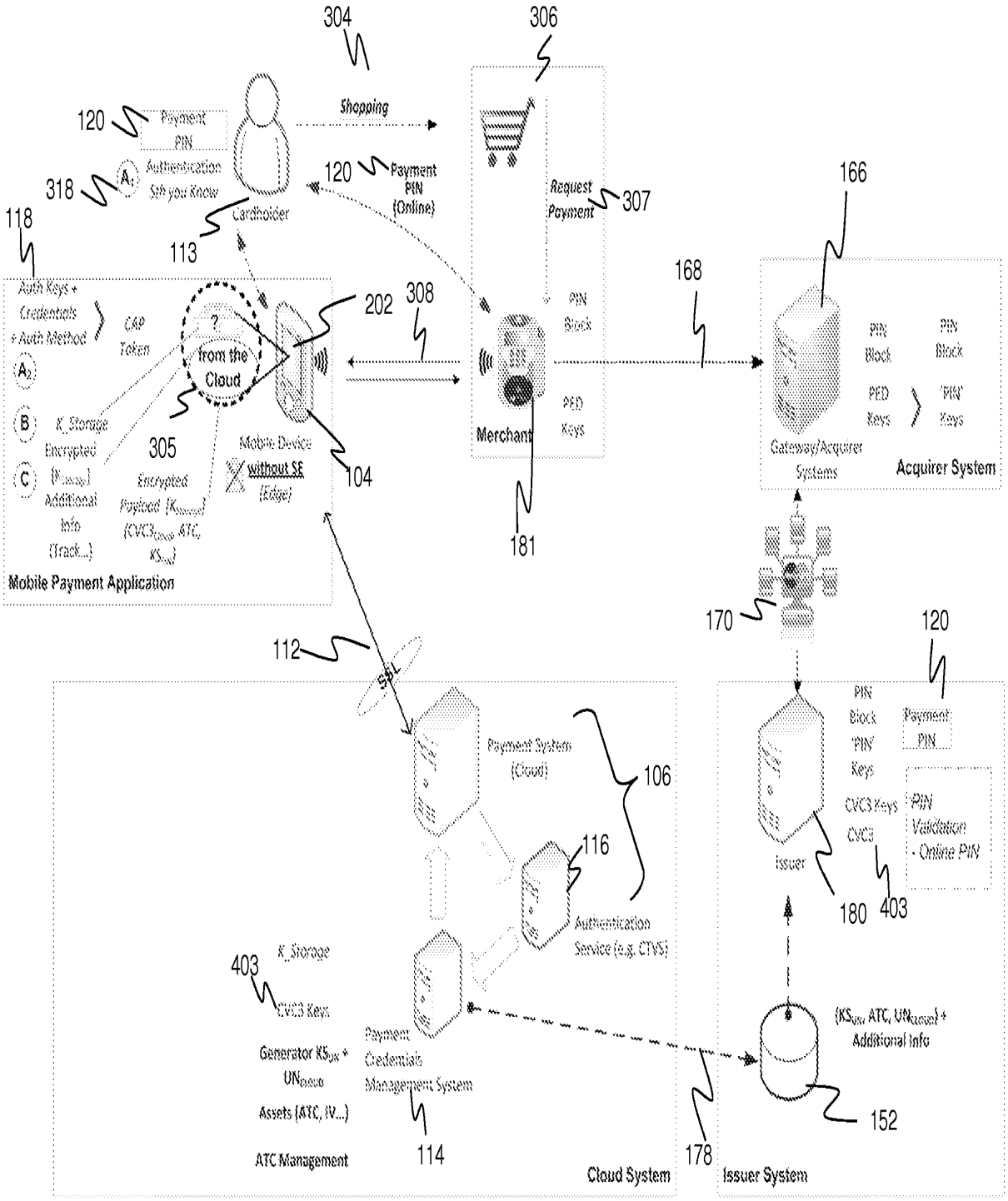

304
306
Shopping
120
Payment PIN
Authentication
Sth you Know
318
Cardholder
113
120
Payment PIN (Online)
Request Payment
307
166
118
Auth Keys + Credentials + Auth Method
CAP Token
from the Cloud
202
308
PIN Block
PED Keys
168
PIN Block
PIN Block
PED Keys
'PIN' Keys
Gateway/Acquirer Systems
Acquirer System
K_Storage
Encrypted
Additional Info (Track...)
305
Encrypted Payload
Mobile Device
without SE (Edge)
104
Merchant
181
Mobile Payment Application
170
120
112
SSL
Payment System (Cloud)
106
116
Authentication Service (e.g. CTVS)
PIN Block
'PIN' Keys
Payment PIN
CVC3 Keys
CVC3
PIN Validation - Online PIN
Issuer
180
403
403
K_Storage
CVC3 Keys
Generator KS_UN + UN_CLOUD
Assets (ATC, IV...)
ATC Management
Payment Credentials Management System
114
Cloud System
178
Issuer System
Additional Info
152

FIG. 4

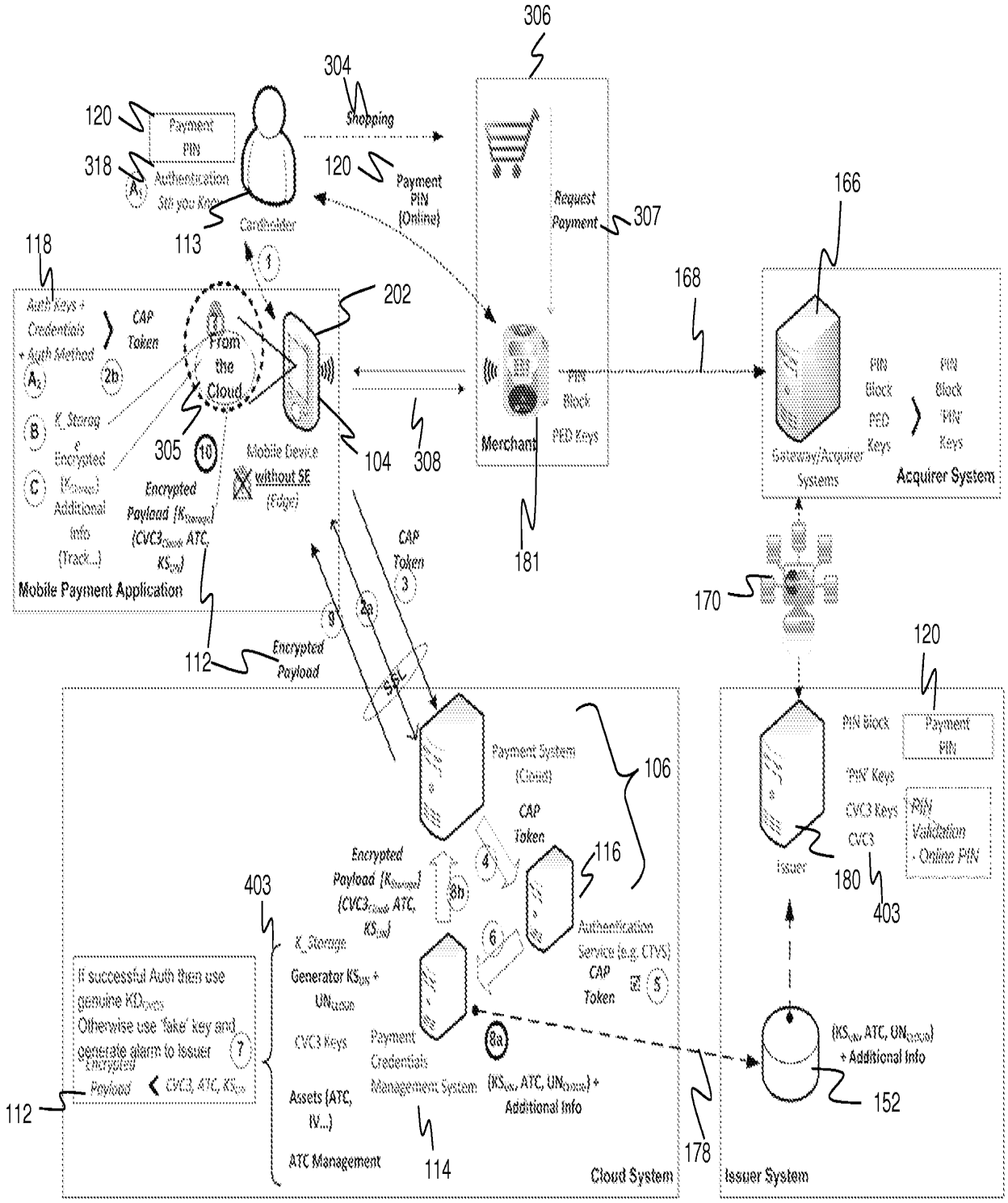

306
304
Shopping
120
Payment PIN
318
Authentication
Still you Know
113
Cardholder
120
Payment PIN (Online)
Request Payment
307
166
118
Auth Keys + Credentials + Auth Method
CAP Token
From the Cloud
202
168
PIN Block
PED Keys
PIN Block
'PIN' Keys
Gateway/Acquirer Systems
Acquirer System
K_Storage
Encrypted
Additional Info (Track...)
305
Mobile Device without SE (Edge)
104
308
Merchant
181
Encrypted Payload
Mobile Payment Application
CAP Token
170
112
Encrypted Payload
SSL
120
Payment System (Cloud)
106
CAP Token
PIN Block
Payment PIN
'PIN' Keys
CVC3 Keys
CVC3
PIN Validation - Online PIN
Issuer
180
403
116
Encrypted Payload
403
K_Storage
Generator
Authentication Service (e.g. CTVS)
CAP Token
If successful Auth then use genuine
Otherwise use 'fake' key and generate alarm to Issuer
CVC3 Keys
Payment Credentials Management System
+ Additional Info
152
Encrypted Payload
112
Assets (ATC, IV...)
ATC Management
114
Cloud System
178
Issuer System

FIG. 5

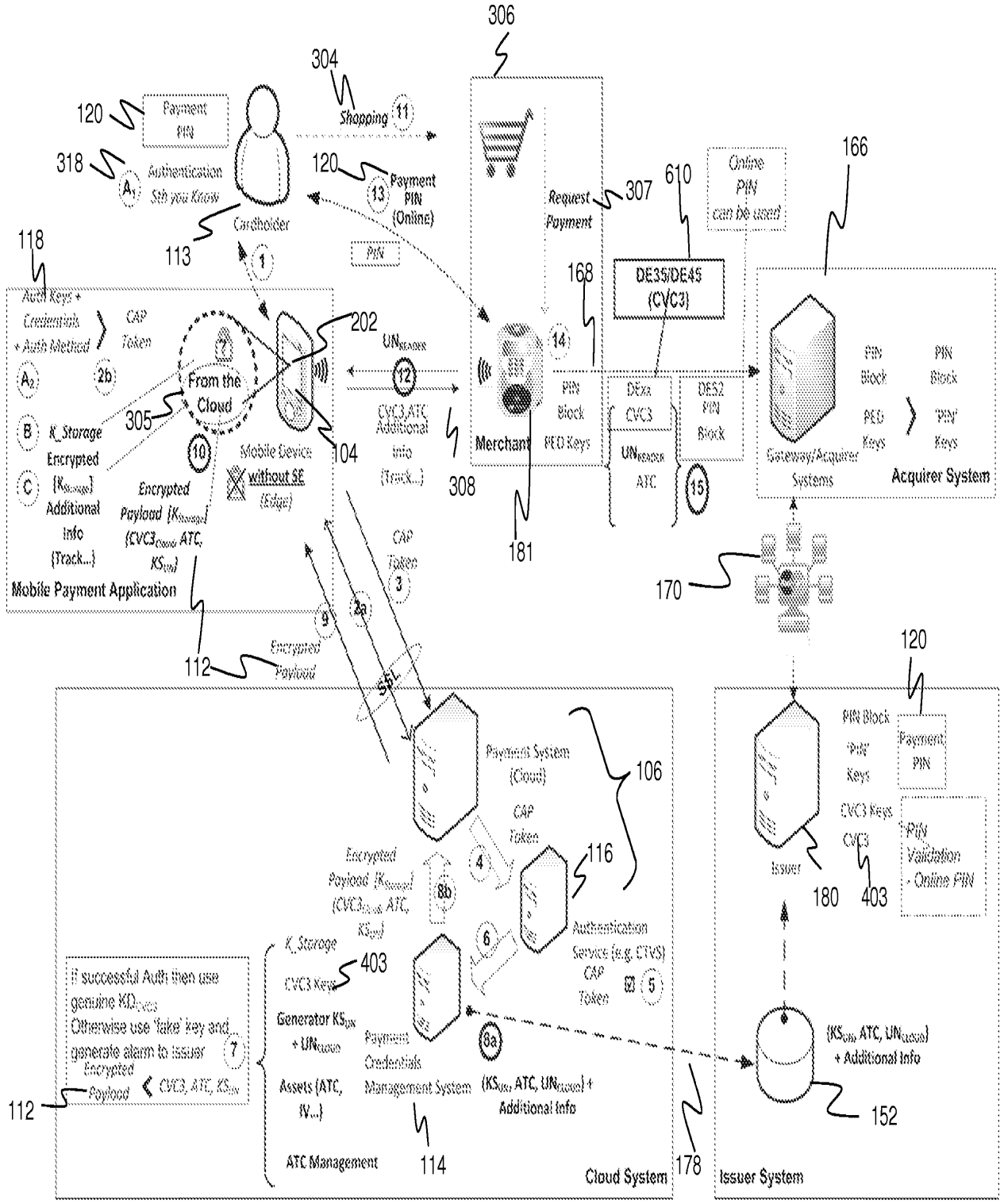
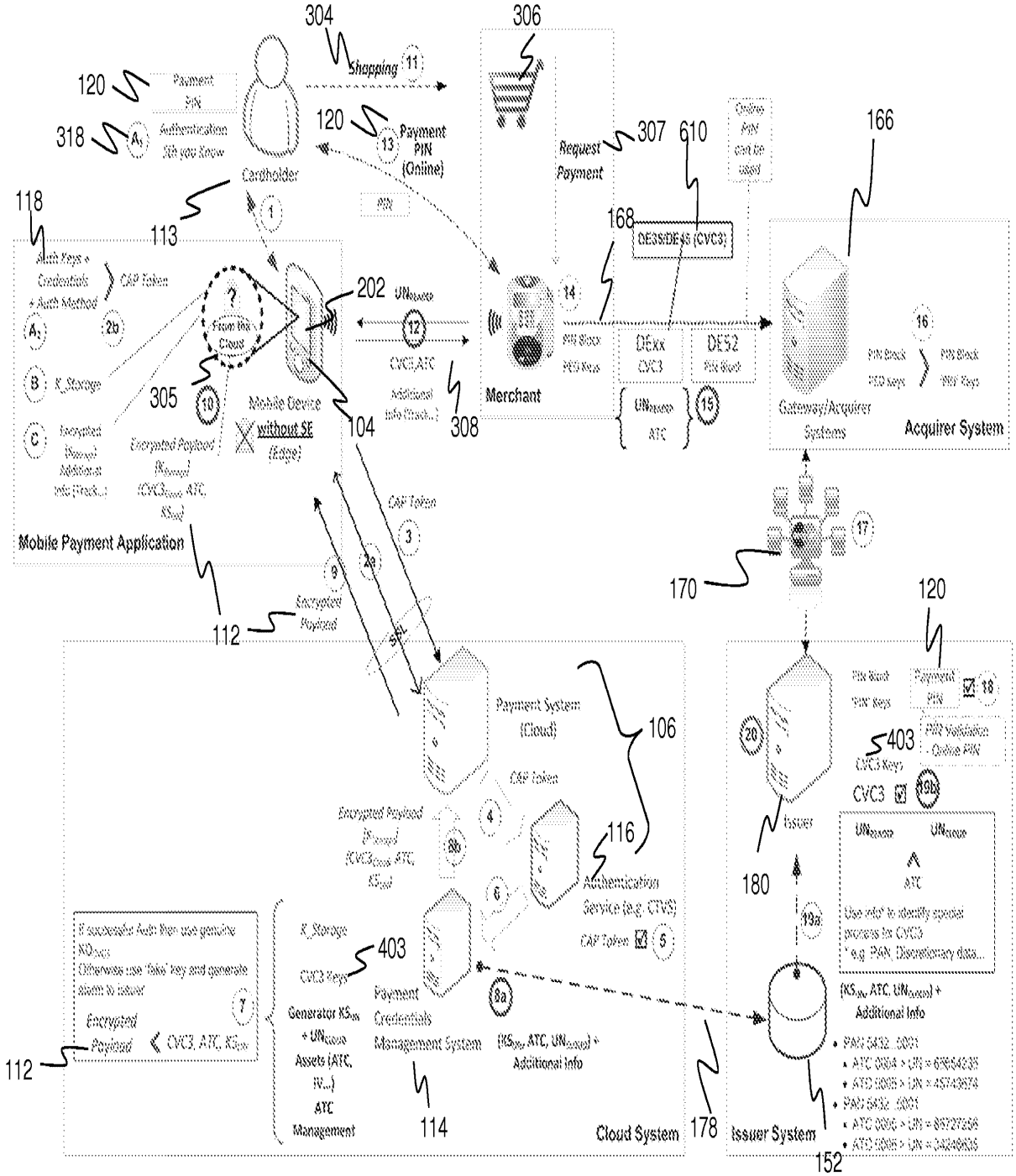

306
304
Shopping
11
120
Payment PIN
318
Authentication Sth you Know
Cardholder
113
120
Payment PIN (Online)
13
PIN
Request Payment
307
610
Online PIN can be used
166
118
Auth Keys + Credentials + Auth Method
CAP Token
2b
From the Cloud
305
202
104
Mobile Device without SE (Edge)
Encrypted Payload
Additional Info (Track...)
Mobile Payment Application
112
CVC3,ATC Additional Info (Track...)
308
168
DE35/DE45 (CVC3)
Merchant
PIN Block
PED Keys
DE52 PIN Block
ATC
15
181
Gateway/Acquirer Systems
PIN Block
PED Keys
'PIN' Keys
Acquirer System
170
CAP Token
3
2a
9
Encrypted Payload
SSL
Payment System (Cloud)
106
CAP Token
116
Encrypted Payload
8b
4
6
Authentication Service (e.g. CTVS)
CAP Token
5
K_Storage
CVC3 Keys
403
Generator
Assets (ATC, IV...)
Payment Credentials Management System
8a
ATC Management
114
Cloud System
178
Issuer
180
PIN Block
'PIN' Keys
Payment PIN
CVC3 Keys
CVC3
403
PIN Validation - Online PIN
Issuer System
152

FIG. 6



304
306
Shopping
120
Payment PIN
318
Authentication 5th you Know
Cardholder
120
Payment PIN (Online)
Request Payment
307
610
166
118
113
168
DE35/DE45 (CVC3)
Auth Keys + Credentials + Auth Method
CAP Token
202
From the Cloud
K_Storage
305
Mobile Device without SE (Edge)
104
308
Merchant
DExx CVC3
DE52
Gateway/Acquirer Systems
Acquirer System
Mobile Payment Application
CAP Token
170
112
Encrypted Payload
120
Payment System (Cloud)
106
CAP Token
Encrypted Payload
116
Authentication Service (e.g. CTVS)
CAP Token
403
Payment Credentials Management System
Generator KS
Assets (ATC, IV...)
ATC Management
Encrypted Payload
CVC3, ATC, KS
112
114
Cloud System
178
Issuer
403
CVC3 Keys
CVC3
180
Issuer System
152

FIG. 7

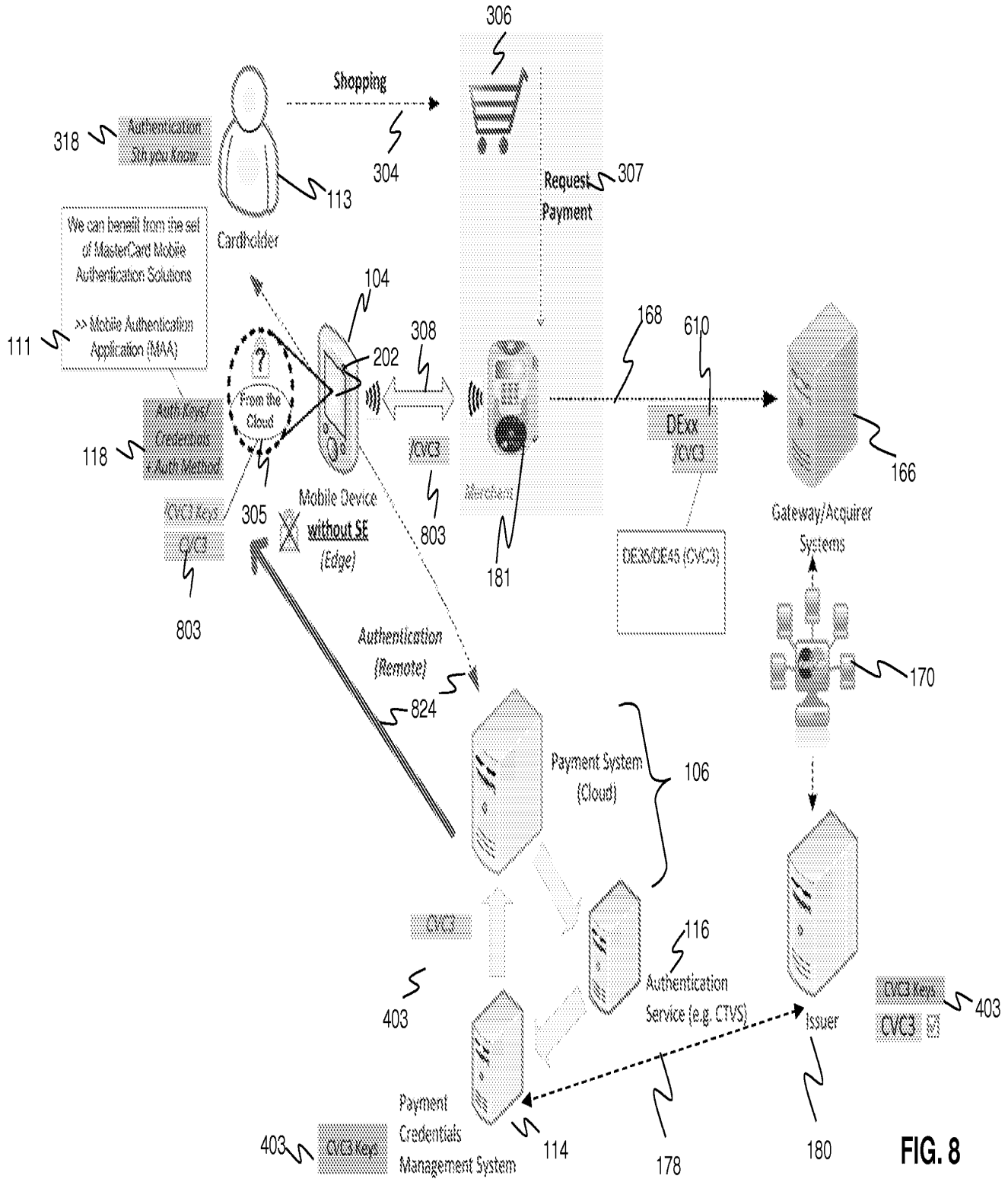
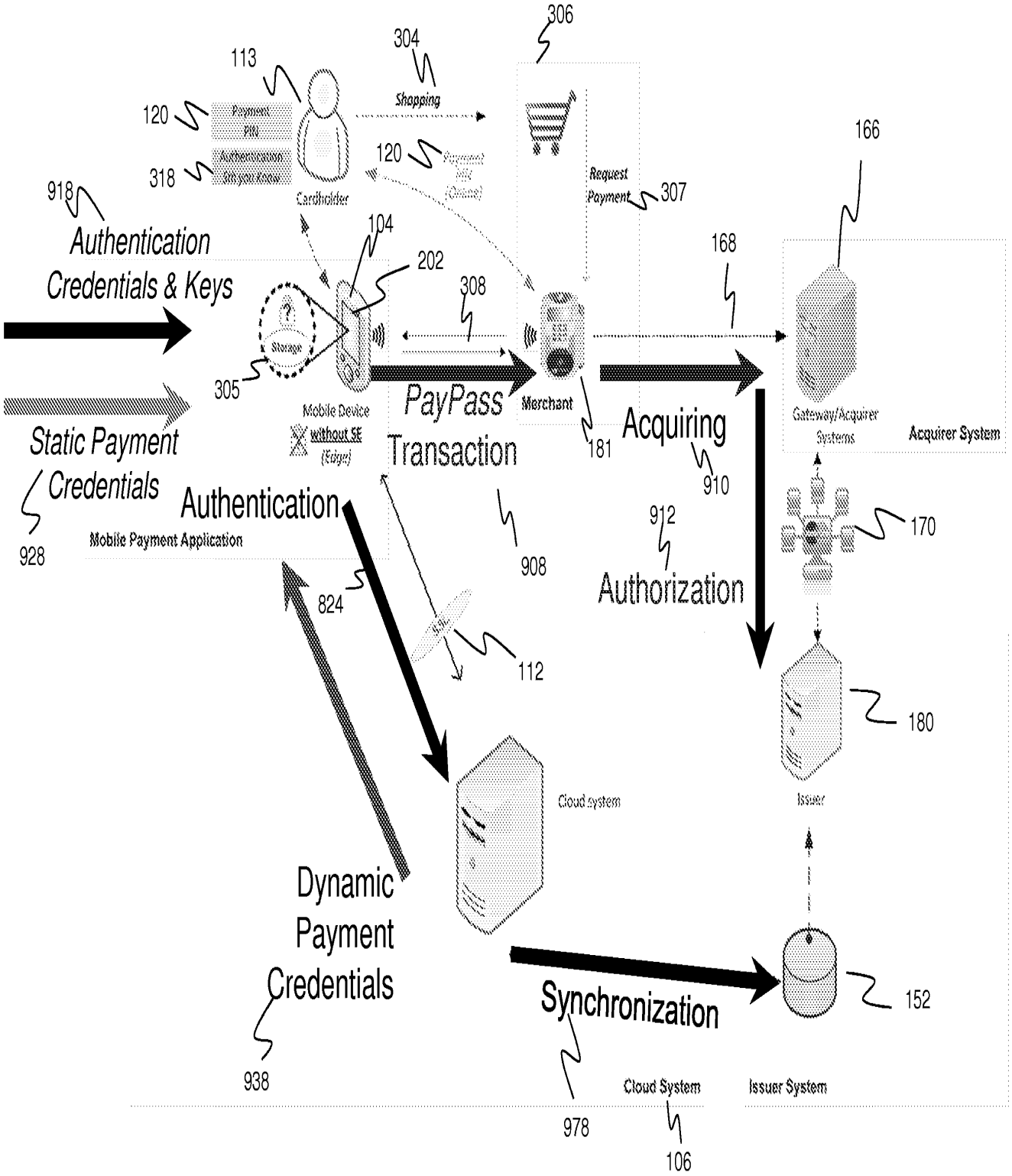
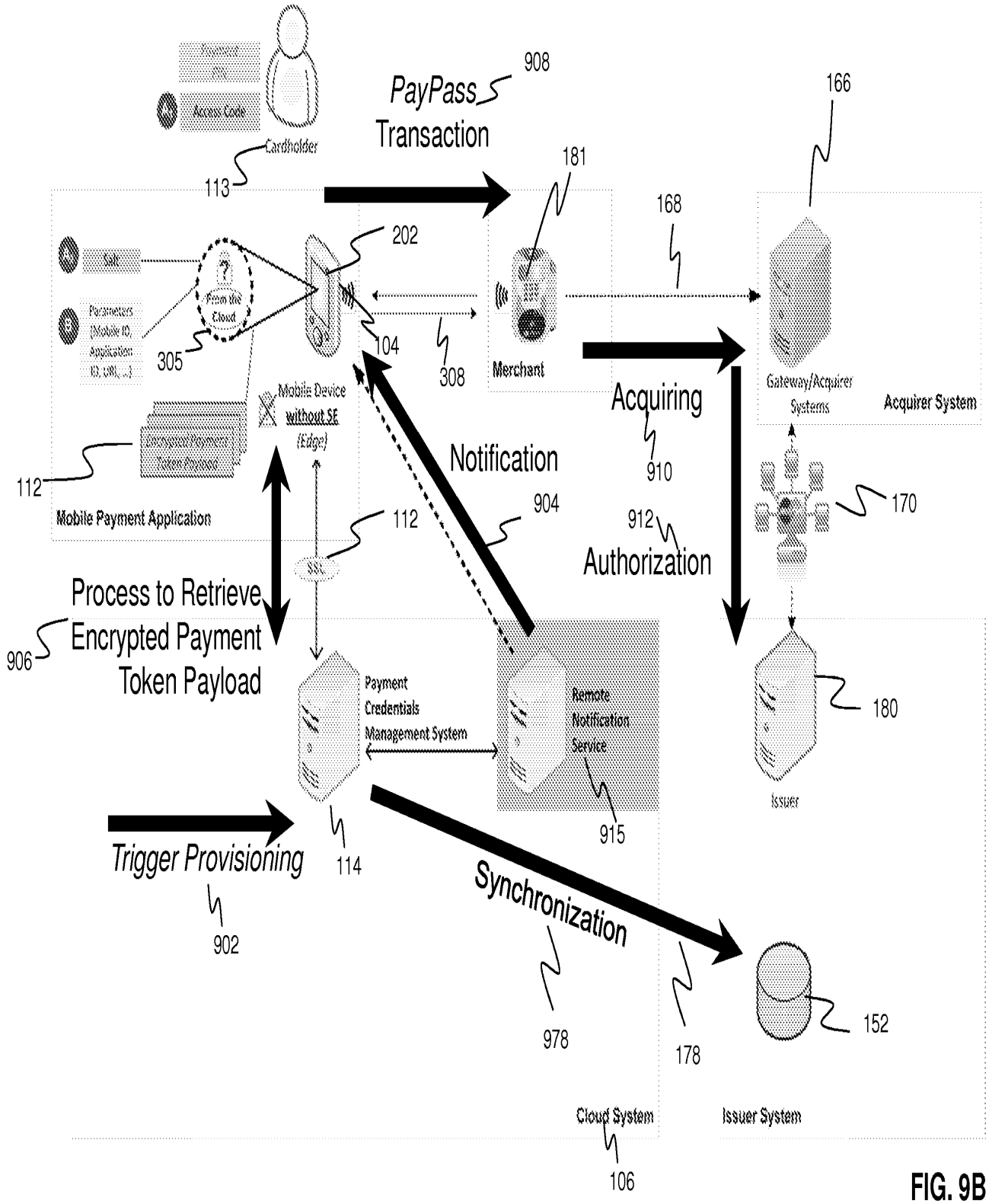

306
Shopping
304
318
Authentication
Sth you Know
113
Cardholder
Request
Payment
307
We can benefit from the set of MasterCard Mobile Authentication Solutions
>> Mobile Authentication Application (MAA)
111
104
202
308
168
610
From the Cloud
Auth Keys/ Credentials + Auth Method
118
/CVC3
DExx
/CVC3
166
Mobile Device
without SE
(Edge)
305
/CVC3 Keys
CVC3
803
Merchant
803
181
DE35/DE45 (CVC3)
Gateway/Acquirer Systems
Authentication
(Remote)
824
170
Payment System
(Cloud)
106
CVC3
116
Authentication
Service (e.g. CTVS)
403
Issuer
CVC3 Keys
CVC3
403
Payment
Credentials
Management System
403
CVC3 Keys
114
178
180

FIG. 8



306
304
113
Shopping
120
Payment PIN
Authentication Sin you Know
318
Cardholder
120
Payment PIN (Online)
Request Payment
307
166
918
Authentication Credentials & Keys
104
202
308
168
Storage
305
Mobile Device
without SE (Edge)
PayPass Transaction
Merchant
181
Acquiring
910
Gateway/Acquirer Systems
Acquirer System
Static Payment Credentials
928
Authentication
Mobile Payment Application
908
912
Authorization
170
824
112
180
Cloud system
Issuer
Dynamic Payment Credentials
938
Synchronization
152
978
Cloud System
Issuer System
106

FIG. 9A



Access Code
Cardholder
113
PayPass
Transaction
908
181
166
168
202
Salt
Parameters (Mobile ID, Application ID, URL, ...)
From the Cloud
305
104
308
Merchant
Gateway/Acquirer Systems
Acquirer System
Mobile Device without SE (Edge)
Encrypted Payment Token Payload
112
Mobile Payment Application
Acquiring
910
Notification
904
112
SSL
912
Authorization
170
Process to Retrieve Encrypted Payment Token Payload
906
Payment Credentials Management System
Remote Notification Service
180
Issuer
915
Trigger Provisioning
114
902
Synchronization
978
178
152
Cloud System
Issuer System
106

FIG. 9B

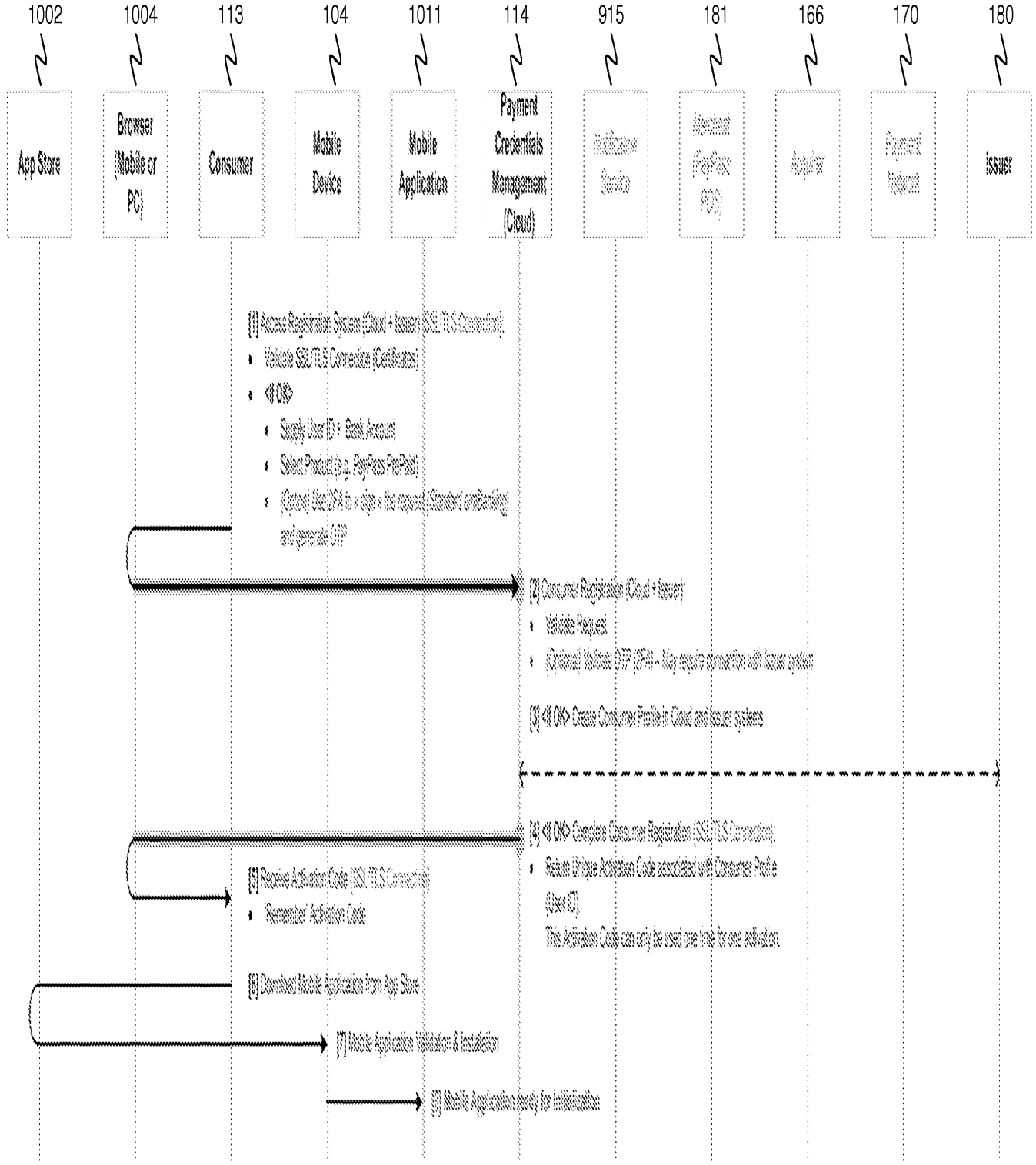

1002
1004
113
104
1011
114
915
181
166
170
180
App Store
Browser (Mobile or PC)
Consumer
Mobile Device
Mobile Application
Payment Credentials Management (Cloud)
Issuer

FIG. 10

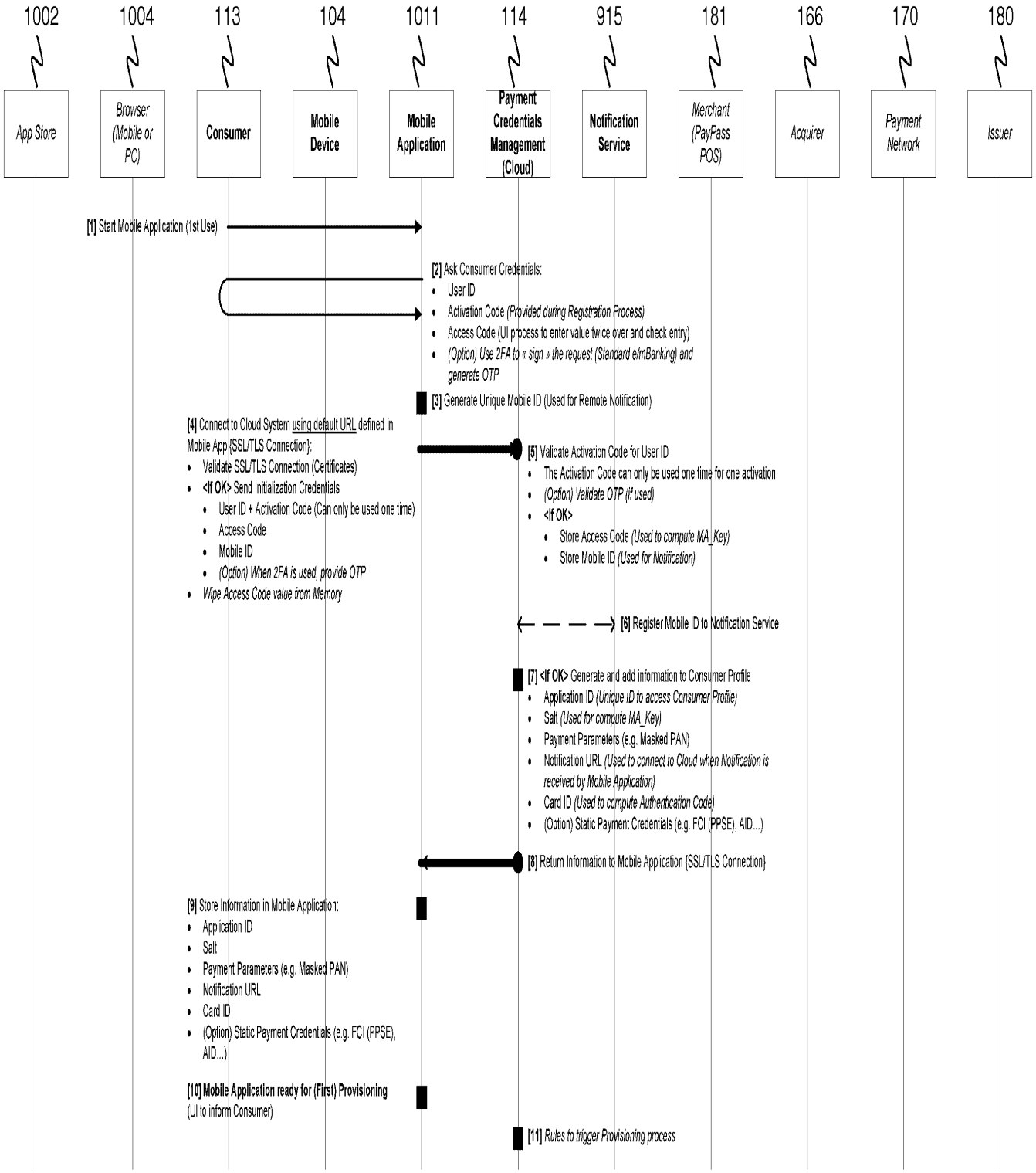

1002
1004
113
104
1011
114
915
181
166
170
180
App Store
Browser (Mobile or PC)
Consumer
Mobile Device
Mobile Application
Payment Credentials Management (Cloud)
Notification Service
Merchant (PayPass POS)
Acquirer
Payment Network
Issuer
[1] Start Mobile Application (1st Use)
[2] Ask Consumer Credentials:
User ID
Activation Code (Provided during Registration Process)
Access Code (UI process to enter value twice over and check entry)
(Option) Use 2FA to « sign » the request (Standard e/mBanking) and generate OTP
[3] Generate Unique Mobile ID (Used for Remote Notification)
[4] Connect to Cloud System using default URL defined in Mobile App (SSL/TLS Connection):
Validate SSL/TLS Connection (Certificates)
<If OK> Send Initialization Credentials
User ID + Activation Code (Can only be used one time)
Access Code
Mobile ID
(Option) When 2FA is used, provide OTP
Wipe Access Code value from Memory
[5] Validate Activation Code for User ID
The Activation Code can only be used one time for one activation.
(Option) Validate OTP (if used)
<If OK>
Store Access Code (Used to compute MA_Key)
Store Mobile ID (Used for Notification)
[6] Register Mobile ID to Notification Service
[7] <If OK> Generate and add information to Consumer Profile
Application ID (Unique ID to access Consumer Profile)
Salt (Used for compute MA_Key)
Payment Parameters (e.g. Masked PAN)
Notification URL (Used to connect to Cloud when Notification is received by Mobile Application)
Card ID (Used to compute Authentication Code)
(Option) Static Payment Credentials (e.g. FCI (PPSE), AID...)
[8] Return Information to Mobile Application (SSL/TLS Connection)
[9] Store Information in Mobile Application:
Application ID
Salt
Payment Parameters (e.g. Masked PAN)
Notification URL
Card ID
(Option) Static Payment Credentials (e.g. FCI (PPSE), AID...)
[10] Mobile Application ready for (First) Provisioning (UI to inform Consumer)
[11] Rules to trigger Provisioning process

FIG. 11

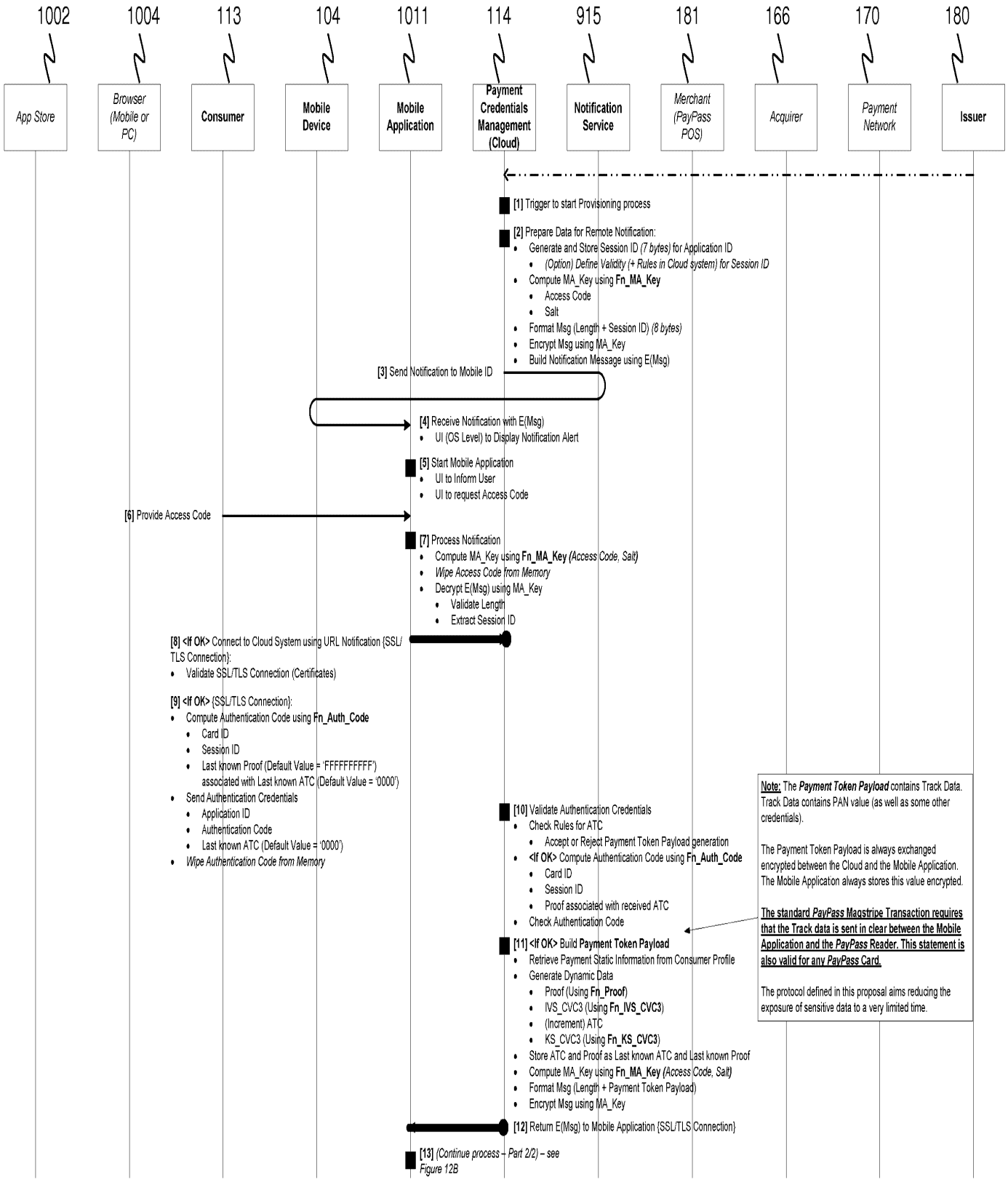

1002
1004
113
104
1011
114
915
181
166
170
180
App Store
Browser (Mobile or PC)
Consumer
Mobile Device
Mobile Application
Payment Credentials Management (Cloud)
Notification Service
Merchant (PayPass POS)
Acquirer
Payment Network
Issuer
[1] Trigger to start Provisioning process
[2] Prepare Data for Remote Notification:
Generate and Store Session ID (7 bytes) for Application ID
(Option) Define Validity (+ Rules in Cloud system) for Session ID
Compute MA_Key using Fn_MA_Key
Access Code
Salt
Format Msg (Length + Session ID) (8 bytes)
Encrypt Msg using MA_Key
Build Notification Message using E(Msg)
[3] Send Notification to Mobile ID
[4] Receive Notification with E(Msg)
UI (OS Level) to Display Notification Alert
[5] Start Mobile Application
UI to Inform User
UI to request Access Code
[6] Provide Access Code
[7] Process Notification
Compute MA_Key using Fn_MA_Key (Access Code, Salt)
Wipe Access Code from Memory
Decrypt E(Msg) using MA_Key
Validate Length
Extract Session ID
[8] <If OK> Connect to Cloud System using URL Notification {SSL/TLS Connection}:
Validate SSL/TLS Connection (Certificates)
[9] <If OK> {SSL/TLS Connection}:
Compute Authentication Code using Fn_Auth_Code
Card ID
Session ID
Last known Proof (Default Value = 'FFFFFFFFFFF') associated with Last known ATC (Default Value = '0000')
Send Authentication Credentials
Application ID
Authentication Code
Last known ATC (Default Value = '0000')
Wipe Authentication Code from Memory
[10] Validate Authentication Credentials
Check Rules for ATC
Accept or Reject Payment Token Payload generation
<If OK> Compute Authentication Code using Fn_Auth_Code
Card ID
Session ID
Proof associated with received ATC
Check Authentication Code
[11] <If OK> Build Payment Token Payload
Retrieve Payment Static Information from Consumer Profile
Generate Dynamic Data
Proof (Using Fn_Proof)
IVS_CVC3 (Using Fn_IVS_CVC3)
(Increment) ATC
KS_CVC3 (Using Fn_KS_CVC3)
Store ATC and Proof as Last known ATC and Last known Proof
Compute MA_Key using Fn_MA_Key (Access Code, Salt)
Format Msg (Length + Payment Token Payload)
Encrypt Msg using MA_Key
[12] Return E(Msg) to Mobile Application {SSL/TLS Connection}
[13] (Continue process – Part 2/2) – see Figure 12B
Note: The Payment Token Payload contains Track Data. Track Data contains PAN value (as well as some other credentials).
The Payment Token Payload is always exchanged encrypted between the Cloud and the Mobile Application. The Mobile Application always stores this value encrypted.
The standard PayPass Magstripe Transaction requires that the Track data is sent in clear between the Mobile Application and the PayPass Reader. This statement is also valid for any PayPass Card.
The protocol defined in this proposal aims reducing the exposure of sensitive data to a very limited time.

FIG. 12A

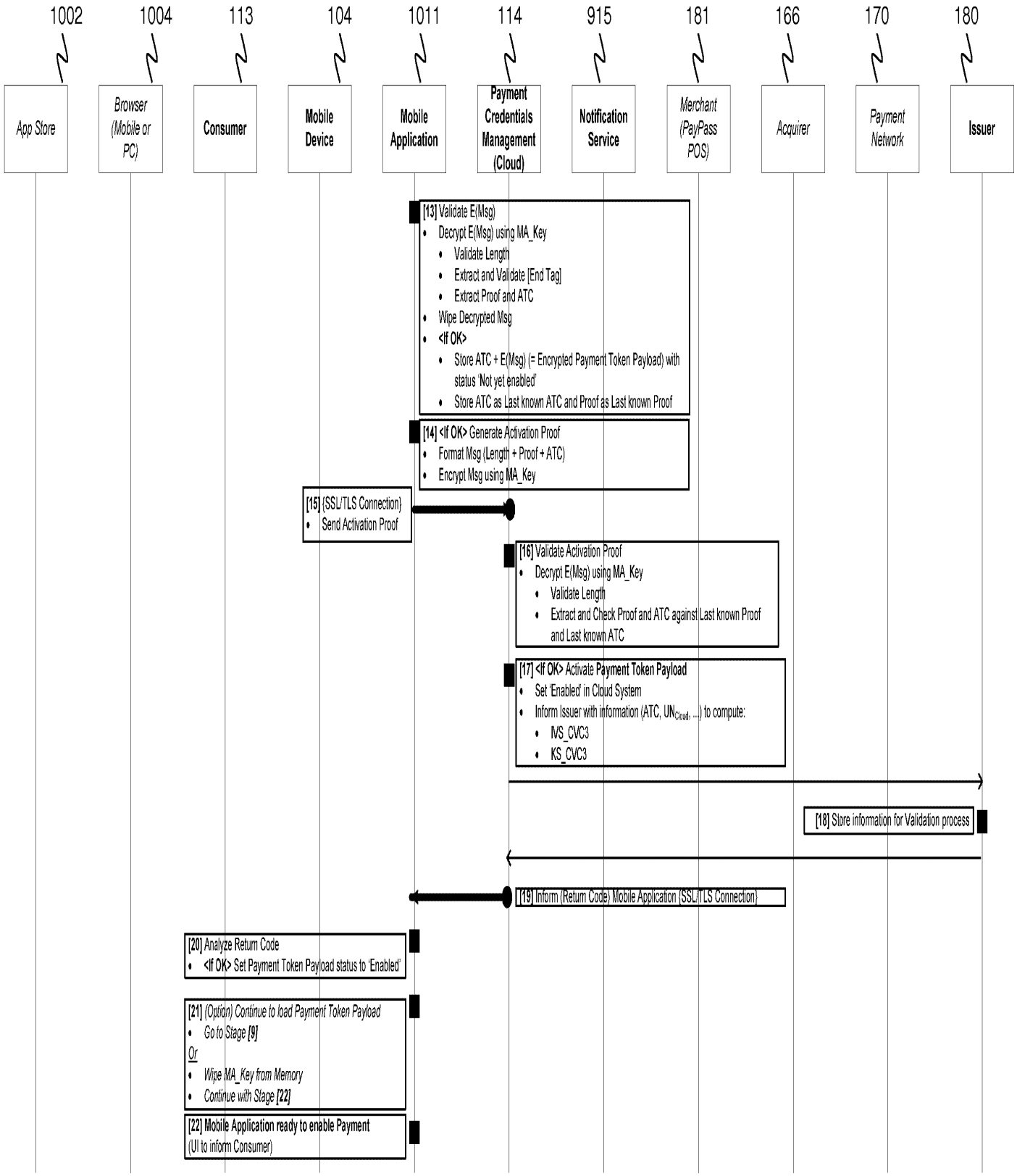
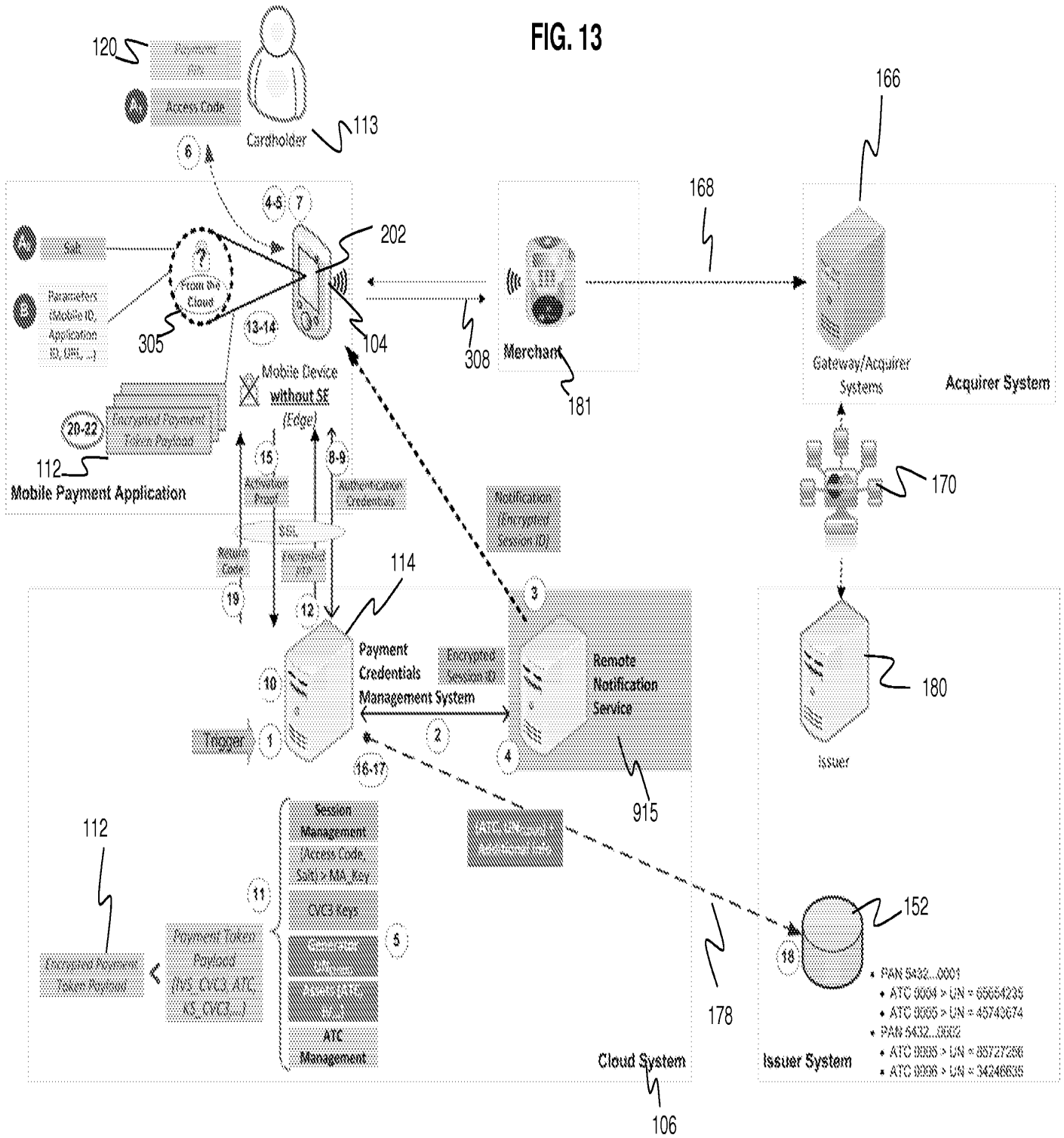

1002
1004
113
104
1011
114
915
181
166
170
180
App Store
Browser (Mobile or PC)
Consumer
Mobile Device
Mobile Application
Payment Credentials Management (Cloud)
Notification Service
Merchant (PayPass POS)
Acquirer
Payment Network
Issuer
[13] Validate E(Msg)
Decrypt E(Msg) using MA_Key
Validate Length
Extract and Validate [End Tag]
Extract Proof and ATC
Wipe Decrypted Msg
<If OK>
Store ATC + E(Msg) (= Encrypted Payment Token Payload) with status 'Not yet enabled'
Store ATC as Last known ATC and Proof as Last known Proof
[14] <If OK> Generate Activation Proof
Format Msg (Length + Proof + ATC)
Encrypt Msg using MA_Key
[15] {SSL/TLS Connection}
Send Activation Proof
[16] Validate Activation Proof
Decrypt E(Msg) using MA_Key
Validate Length
Extract and Check Proof and ATC against Last known Proof and Last known ATC
[17] <If OK> Activate Payment Token Payload
Set 'Enabled' in Cloud System
Inform Issuer with information (ATC, UN_Cloud, ...) to compute:
IVS_CVC3
KS_CVC3
[18] Store information for Validation process
[19] Inform (Return Code) Mobile Application {SSL/TLS Connection}
[20] Analyze Return Code
<If OK> Set Payment Token Payload status to 'Enabled'
[21] (Option) Continue to load Payment Token Payload
Go to Stage [9]
Or
Wipe MA_Key from Memory
Continue with Stage [22]
[22] Mobile Application ready to enable Payment
(UI to inform Consumer)

FIG. 12B

FIG. 13



120
Access Code
Cardholder
113
6
4-5
7
202
Salt
Parameters (Mobile ID, Application ID, URL, ...)
From the Cloud
305
13-14
104
308
Merchant
181
168
166
Gateway/Acquirer Systems
Acquirer System
Mobile Device without SE (Edge)
20-22
Encrypted Payment Token Payload
112
Mobile Payment Application
15
8-9
Activation Proof
Authentication Credentials
SSL
Return Code
Encrypted PIN
19
12
Notification (Encrypted Session ID)
170
114
3
Payment Credentials Management System
Encrypted Session ID
Remote Notification Service
10
Trigger
1
2
4
16-17
915
180
Issuer
112
Session Management
(Access Code, Salt) > MA_Key
11
CVC3 Keys
5
Encrypted Payment Token Payload
Payment Token Payload (IVS, CVC3, ATC, KS_CVC3,...)
ATC Management
Cloud System
106
178
152
18
Issuer System

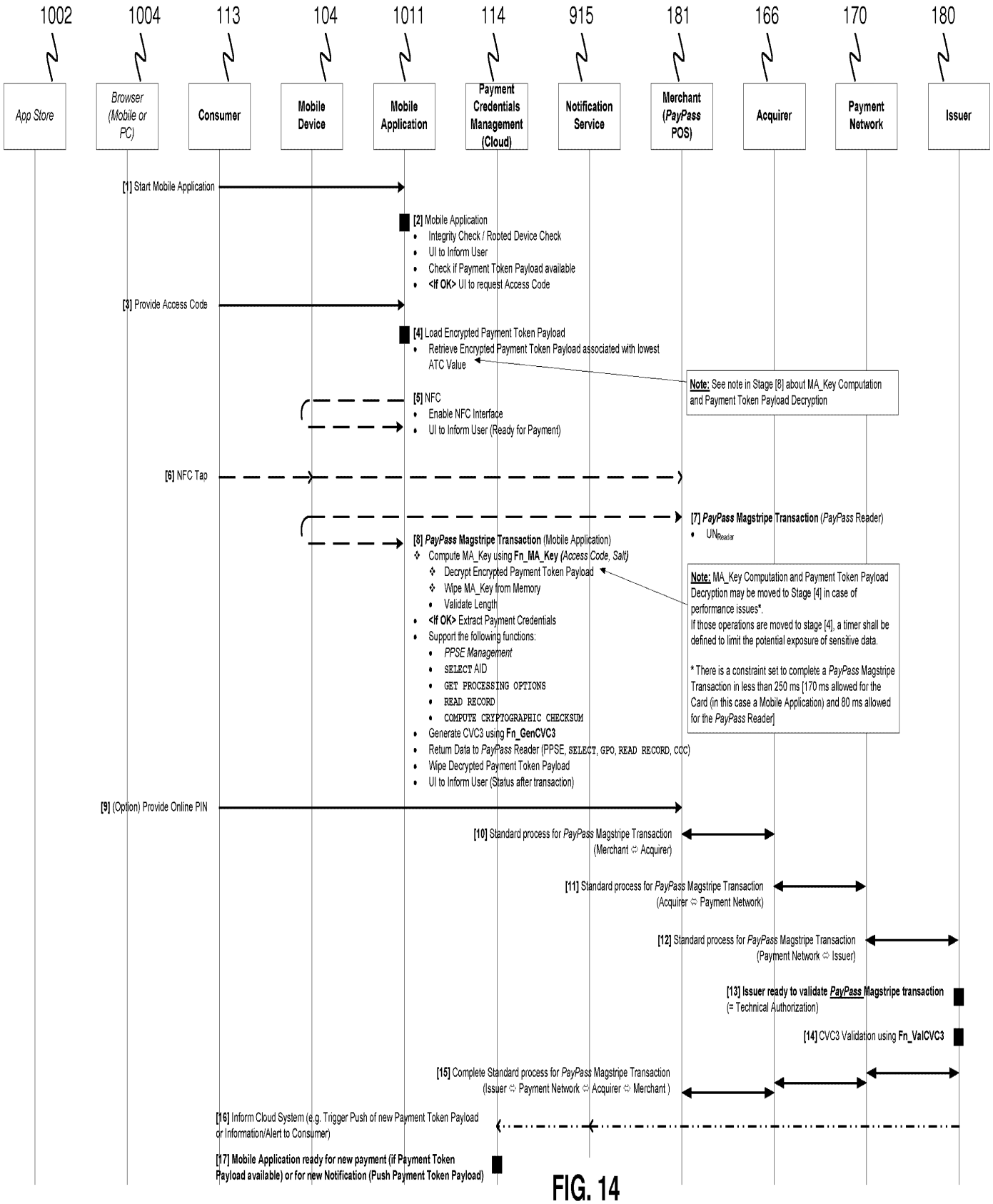

1002
1004
113
104
1011
114
915
181
166
170
180
App Store
Browser (Mobile or PC)
Consumer
Mobile Device
Mobile Application
Payment Credentials Management (Cloud)
Notification Service
Merchant (PayPass POS)
Acquirer
Payment Network
Issuer
[1] Start Mobile Application
[2] Mobile Application
Integrity Check / Rooted Device Check
UI to Inform User
Check if Payment Token Payload available
<If OK> UI to request Access Code
[3] Provide Access Code
[4] Load Encrypted Payment Token Payload
Retrieve Encrypted Payment Token Payload associated with lowest ATC Value
Note: See note in Stage [8] about MA_Key Computation and Payment Token Payload Decryption
[5] NFC
Enable NFC Interface
UI to Inform User (Ready for Payment)
[6] NFC Tap
[7] PayPass Magstripe Transaction (PayPass Reader)
UN_Reader
[8] PayPass Magstripe Transaction (Mobile Application)
Compute MA_Key using Fn_MA_Key (Access Code, Salt)
Decrypt Encrypted Payment Token Payload
Wipe MA_Key from Memory
Validate Length
<If OK> Extract Payment Credentials
Support the following functions:
PPSE Management
SELECT AID
GET PROCESSING OPTIONS
READ RECORD
COMPUTE CRYPTOGRAPHIC CHECKSUM
Generate CVC3 using Fn_GenCVC3
Return Data to PayPass Reader (PPSE, SELECT, GPO, READ RECORD, CCC)
Wipe Decrypted Payment Token Payload
UI to Inform User (Status after transaction)
Note: MA_Key Computation and Payment Token Payload Decryption may be moved to Stage [4] in case of performance issues*.
If those operations are moved to stage [4], a timer shall be defined to limit the potential exposure of sensitive data.
* There is a constraint set to complete a PayPass Magstripe Transaction in less than 250 ms [170 ms allowed for the Card (in this case a Mobile Application) and 80 ms allowed for the PayPass Reader]
[9] (Option) Provide Online PIN
[10] Standard process for PayPass Magstripe Transaction (Merchant ⇔ Acquirer)
[11] Standard process for PayPass Magstripe Transaction (Acquirer ⇔ Payment Network)
[12] Standard process for PayPass Magstripe Transaction (Payment Network ⇔ Issuer)
[13] Issuer ready to validate PayPass Magstripe transaction (= Technical Authorization)
[14] CVC3 Validation using Fn_ValCVC3
[15] Complete Standard process for PayPass Magstripe Transaction (Issuer ⇔ Payment Network ⇔ Acquirer ⇔ Merchant)
[16] Inform Cloud System (e.g. Trigger Push of new Payment Token Payload or Information/Alert to Consumer)
[17] Mobile Application ready for new payment (if Payment Token Payload available) or for new Notification (Push Payment Token Payload)

FIG. 14

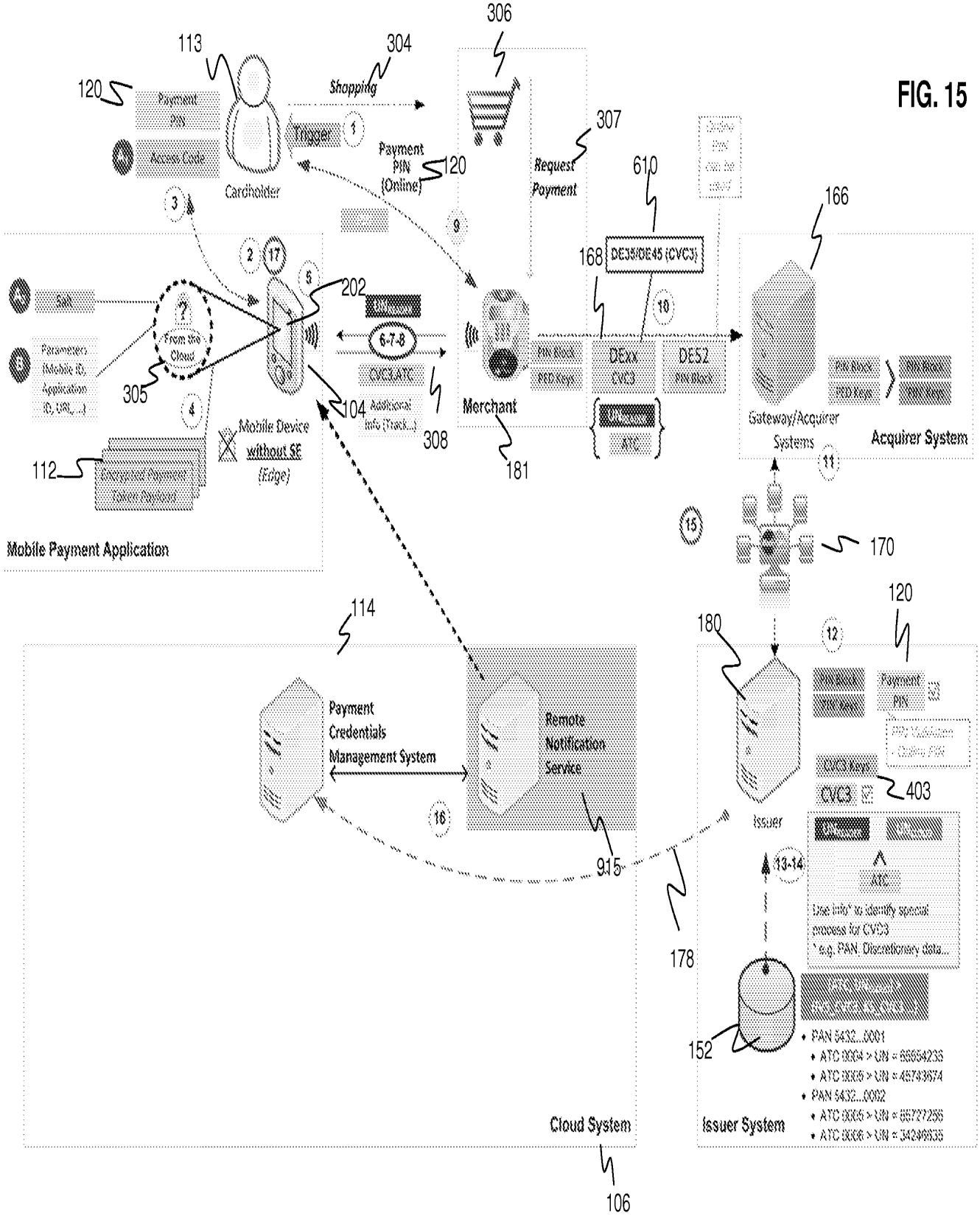


FIG. 15
Cardholder
Shopping
Trigger
Payment PIN (Online)
Request Payment
Merchant
Mobile Device without SE (Edge)
Mobile Payment Application
Gateway/Acquirer Systems
Acquirer System
Payment Credentials Management System
Remote Notification Service
Issuer
Cloud System
Issuer System

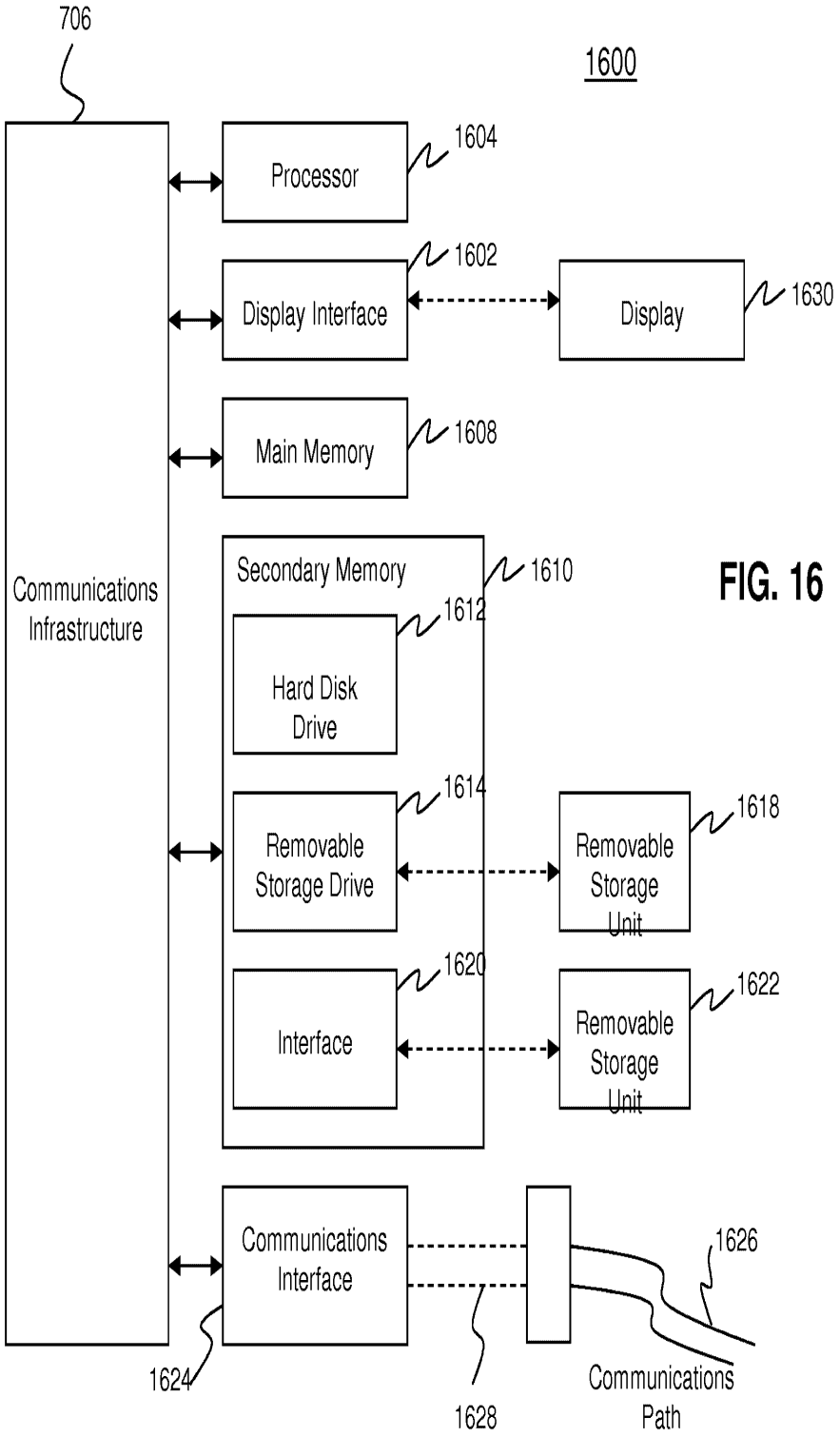
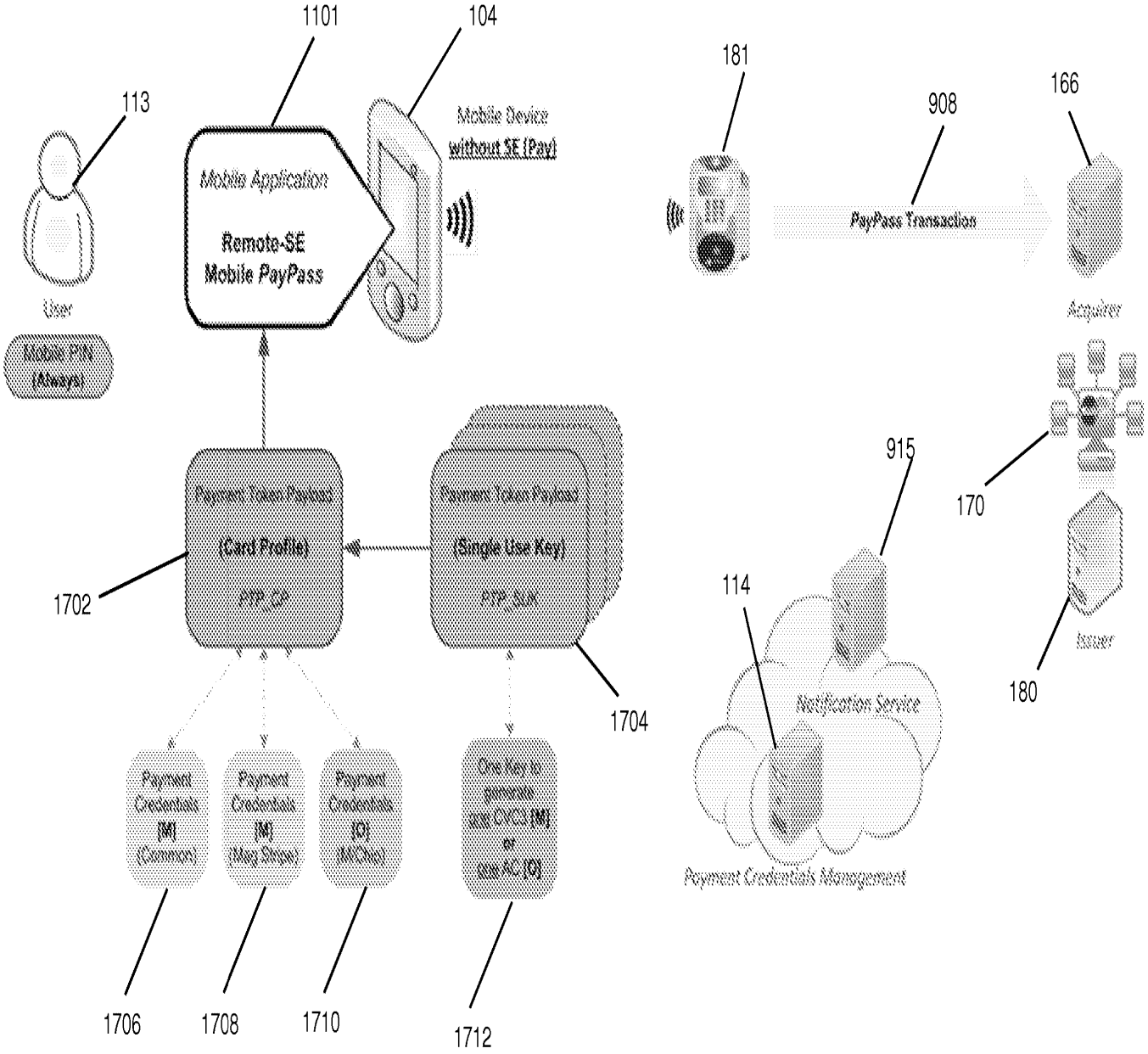

706
1600
Processor
1604
Display Interface
1602
Display
1630
Main Memory
1608
Communications Infrastructure
Secondary Memory
1610
1612
Hard Disk Drive
1614
Removable Storage Drive
Removable Storage Unit
1618
1620
Interface
Removable Storage Unit
1622
Communications Interface
1626
1624
1628
Communications Path

FIG. 16

FIG. 17


1101
104
113
Mobile Device
without SE (Pay)
Mobile Application
Remote-SE
Mobile PayPass
User
Mobile PIN
(Always)
181
908
166
PayPass Transaction
Acquirer
Payment Token Payload
(Card Profile)
Payment Token Payload
(Single Use Key)
1702
1704
915
170
114
Issuer
180
Notification Service
Payment Credentials
[M]
(Common)
Payment Credentials
[M]
(Mag Stripe)
Payment Credentials
[O]
(M/Chip)
One Key to
generate
one CVC3 [M]
or
one AC [O]
Payment Credentials Management
1706
1708
1710
1712

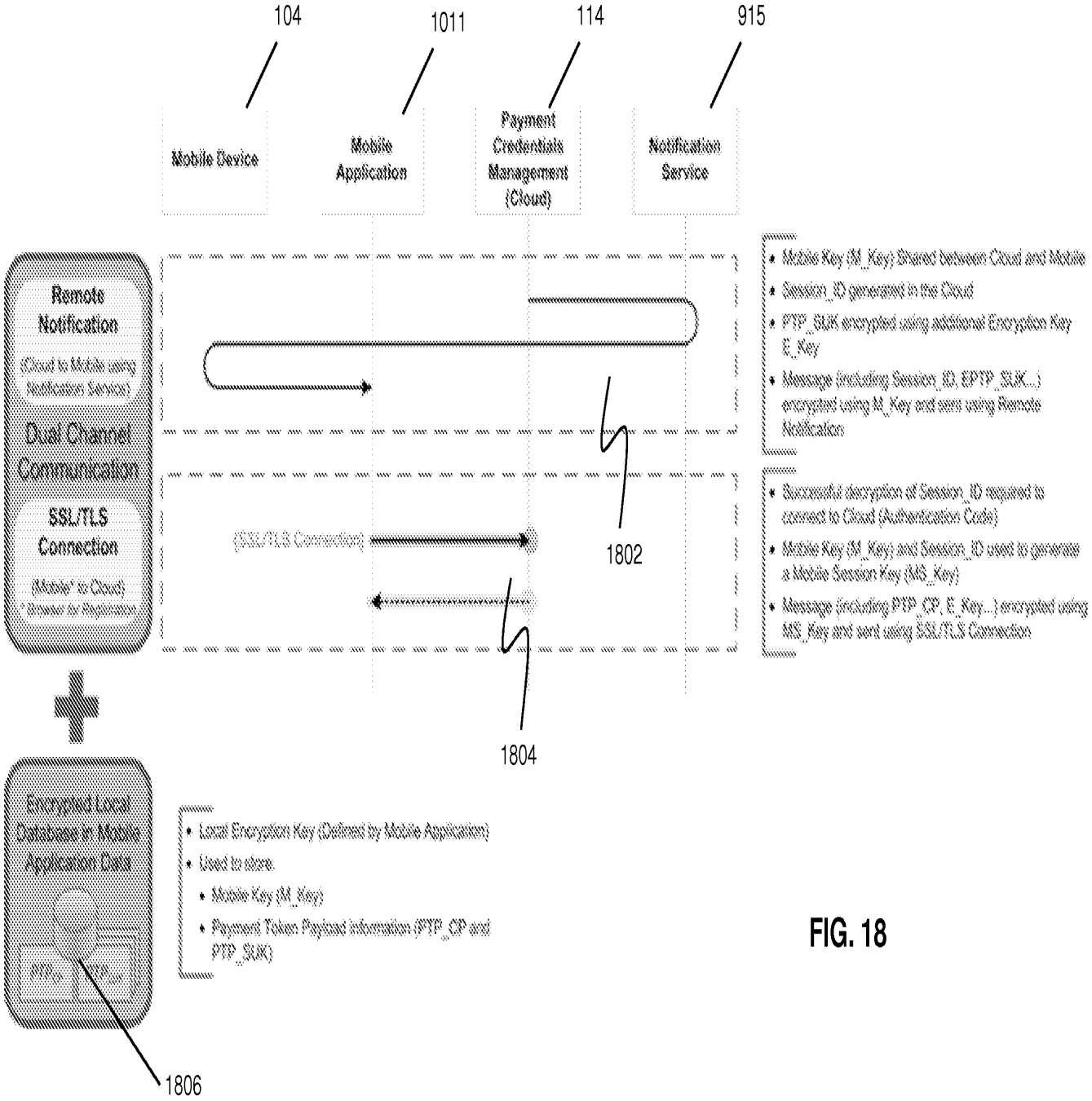
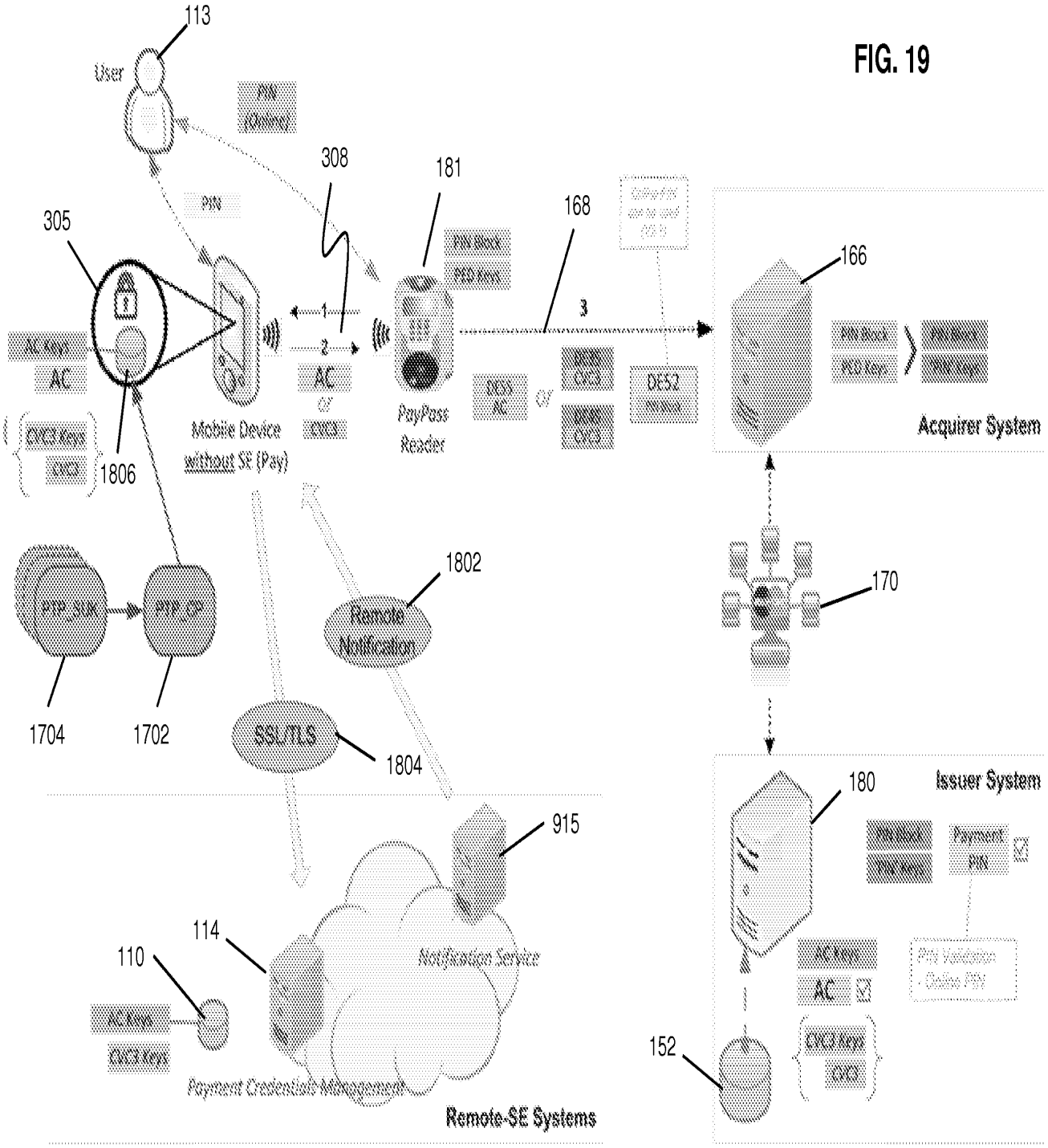

104
1011
114
915
Mobile Device
Mobile Application
Payment Credentials Management (Cloud)
Notification Service
Remote Notification
(Cloud to Mobile using Notification Service)
Dual Channel Communication
SSL/TLS Connection
(Mobile* to Cloud)
* Browser for Registration
(SSL/TLS Connection)
1802
1804
Mobile Key (M_Key) Shared between Cloud and Mobile
Session_ID generated in the Cloud
PTP_SUK encrypted using additional Encryption Key E_Key
Message (including Session_ID, EPTP_SUK...) encrypted using M_Key and sent using Remote Notification
Successful decryption of Session_ID required to connect to Cloud (Authentication Code)
Mobile Key (M_Key) and Session_ID used to generate a Mobile Session Key (MS_Key)
Message (including PTP_CP, E_Key...) encrypted using MS_Key and sent using SSL/TLS Connection
Encrypted Local Database in Mobile Application Data
Local Encryption Key (Defined by Mobile Application)
Used to store:
Mobile Key (M_Key)
Payment Token Payload Information (PTP_CP and PTP_SUK)
1806

FIG. 18

FIG. 19



113
User
PIN (Online)
PIN
308
181
168
305
PIN Block
PED Keys
166
AC Keys
AC
1
2
3
AC
OT
CVC3
PayPass Reader
DES5 AC
OT
DES2
PIN Block
PED Keys
PIN Block
'PIN' Keys
Acquirer System
CVC3 Keys
CVC3
1806
Mobile Device without SE (Pay)
1802
Remote Notification
170
PTP_SUK
PTP_CP
1704
1702
SSL/TLS
1804
Issuer System
180
915
PIN Block
'PIN' Keys
Payment PIN
114
110
Notification Service
AC Keys
AC
PIN Validation - Online PIN
AC Keys
CVC3 Keys
CVC3 Keys
CVC3
152
Payment Credentials Management
Remote-SE Systems

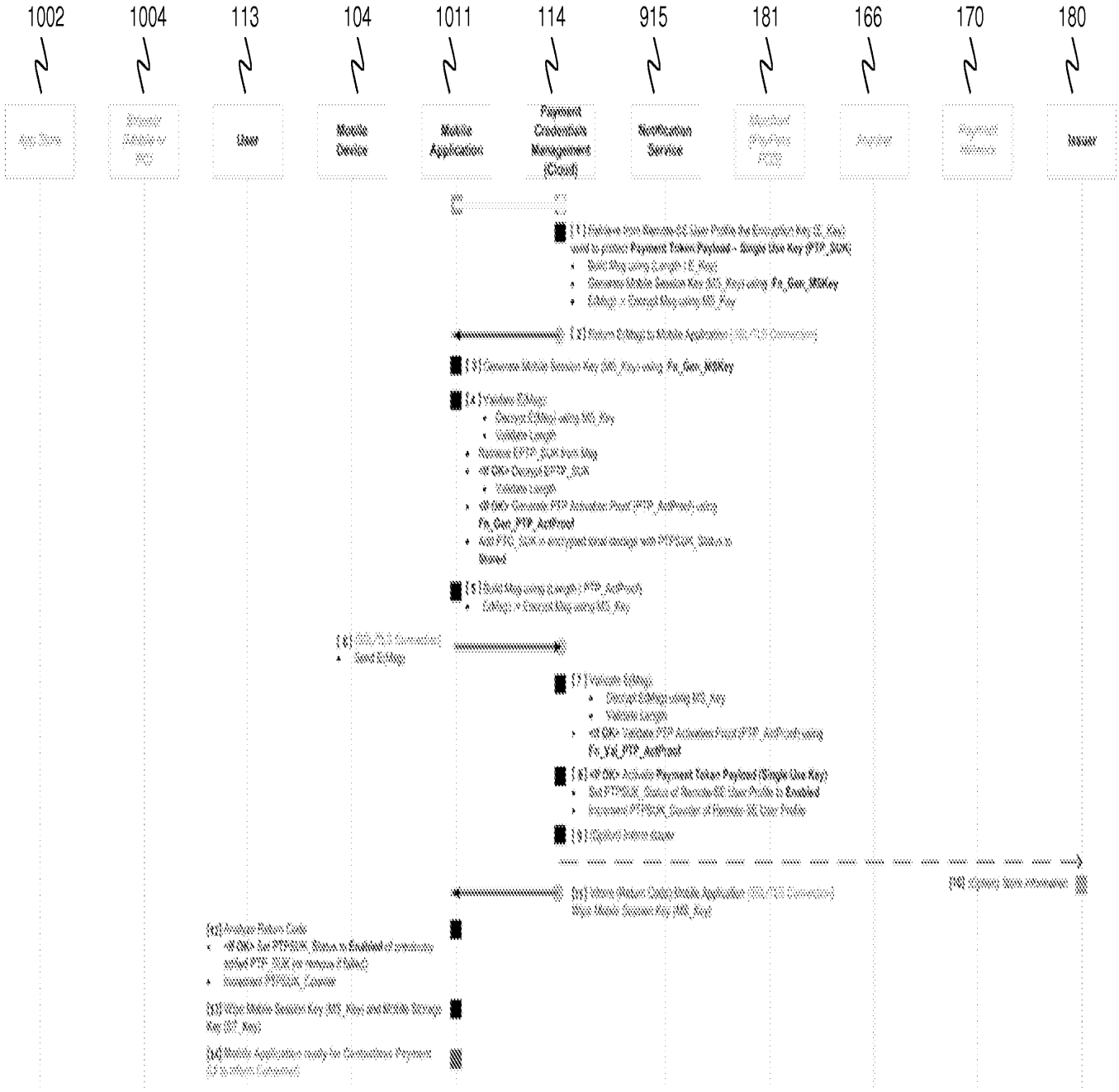

1002
1004
113
104
1011
114
915
181
166
170
180
User
Mobile Device
Mobile Application
Payment Credentials Management (Cloud)
Notification Service
Issuer

FIG. 20