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 |
|---|---|---|---|---|---|---|
| 61/635,248 | 04/18/2012 | | 250 | 0076412-000069 | | |

**CONFIRMATION NO. 3441**

**FILING RECEIPT**

*OC000000053923483*

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

Date Mailed: 04/27/2012

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**

**Applicant(s)**

Simon E.J. PHILLIPS, York, UNITED KINGDOM;
Susan THOMPSON, Nantwich, UNITED KINGDOM;
Mehdi COLLINGE, Braine-l'Alleud, BELGIUM;
David Anthony ROBERTS, Warrington, UNITED KINGDOM;

**Power of Attorney:**
Charles Wieland III--33096

**If Required, Foreign Filing License Granted:** 04/25/2012
The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 61/635,248**
**Projected Publication Date:** None, application is not eligible for pre-grant publication
**Non-Publication Request:** No
**Early Publication Request:** No
**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 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).

---

# APPLICATION FOR
# UNITED STATES LETTERS PATENT

# FOR

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

**By**

**Simon Phillips**
**Susan Thompson**
**David Anthony Roberts**
**and**
**Mehdi Collinge**

**Attorney Docket No. 0076412-000069**
**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

## FIELD OF THE DISCLOSURE

**[0001]** 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

**[0002]** 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.

**[0003]** 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

**[0004]** 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.

**[0005]** 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.

**[0006]** 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.

**[0007]** 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.

**[0008]** 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.

**[0009]** 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.

**[0010]** 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).

**[0011]** In order to prevent cloning of camouflaged data, an embodiment uses a unique Mobile Device ID as a parameter along with the Personal Value.

**[0012]** 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.

**[0013]** 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).

**[0014]** 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.

**[0015]** 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

**[0016]** 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.

**[0017]** 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.

**[0018]** 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.

**[0019]** 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.

**[0020]** Figure 5 illustrates a process flow for provisioning payment credentials for a contactless payment, in accordance with an exemplary embodiment of the present disclosure.

**[0021]** Figure 6 illustrates a process flow for processing a contactless payment transaction, in accordance with an exemplary embodiment of the present disclosure.

**[0022]** Figure 7 illustrates a process flow for approving a contactless payment transaction, in accordance with an exemplary embodiment of the present disclosure.

**[0023]** 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.

**[0024]** 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.

**[0025]** 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.

**[0026]** 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.

**[0027]** 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.

**[0028]** 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.

**[0029]** 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.

**[0030]** 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.

**[0031]** 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.

**[0032]** Figure 16 depicts an example computer system in which embodiments of the present invention may be implemented.

**[0033]** 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

**[0034]** 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.

**[0035]** 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.

**[0036]** 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.

**[0037]** 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.

**[0038]** 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

**[0039]** 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.

**[0040]** 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.

**[0041]** 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.

**[0042]** 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).

**[0043]** 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).

**[0044]** 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).

**[0045]** 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.

**[0046]** 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).

**[0047]** 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 113. In the exemplary embodiment of Figure 1, the encrypted information 113 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.

**[0048]** 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.

**[0049]** 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 $UN_{Reader}$.

**[0050]** The acquirer 166 then routes authorization request 168 to a payment processing network 170 associated with the payment processor 103 (e.g., MasterCard).

**[0051]** 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.

**[0052]** 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.

**[0053]** 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).

**[0054]** 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.

**[0055]** 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

**[0056]** 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

**[0057]** 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.

**[0058]** 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.

**[0059]** With continued reference to Figure 2A, in step 203, the mobile payment application is downloaded and installed.

**[0060]** In step 205, authentication credentials associated with a payment card are provisioned to the mobile device 104.

**[0061]** 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.

**[0062]** 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).

**[0063]** 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**

**[0064]** 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.

**[0065]** 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).

**[0066]** 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.

**[0067]** 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).

**[0068]** 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.

**[0069]** With continued reference to Figure 2B, in step 203, the mobile payment application is downloaded and installed.

**[0070]** 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.

**[0071]** 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.

**[0072]** 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.

**[0073]** 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.

**[0074]** 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.

**[0075]** 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**

**[0076]** 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**

**[0077]** 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**

**[0078]** 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

**[0079]** 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

**[0080]** 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.

**[0081]** As shown in Figure 3, authentication to the cloud-based transaction data generation system 106 is performed to retrieve the payment credentials 174.

**[0082]** 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.

**[0083]** 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.

**[0084]** 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:<br>• The Access to the Mobile Device (e.g. Device Locking mechanism)<br>• The Access to the Mobile Payment Application<br>• 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 |
| **4** | The Payment System (in the Cloud) validates the CAP Token using a CAP Token Validation Service (CTVS).<br>The Payment System can be operated by MasterCard or by the Issuer. |
| **5** | The CTVS validates the CAP Token.<br>The CTVS can be operated by MasterCard or by the Issuer. |

**Table 1**

| **Reference** | **Description** |
|---|---|
| **6** | The result of the CAP Token validation is sent to the Payment Credentials Management System.<br>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.<br>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. |
| **10** | ($KS_{UN}$, $Cloud\_CVC3_{TRACK1/2}$, ATC) and the additional assets are stored.<br>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.<br>The Mobile Payment Application use a specific process to generate the CVC3 using ($KS_{UN}$, $Cloud\_CVC3_{TRACK1/2}$, ATC and $UN_{READER}$) |

**Table 1**

| **Reference** | **Description** |
|---|---|
| **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) |
| **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.<br>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.<br>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. |

**[0085]** 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.

**[0086]** 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.

**[0087]** 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.

**[0088]** 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.

**[0089]** 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.

**[0090]** 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.

**[0091]** 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.

**[0092]** 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).

**[0093]** 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.

**[0094]** 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

**$KD_{CVC3}$**

**[0095]** 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:

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

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

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

<u>(Static) Information known by the Mobile Payment Application</u>

- ◊ 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
    - $PCVC3_{TRACK1/2}$
    - $PUNATC_{TRACK1/2}$
    - $NATC_{TRACK1/2}$

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

- ◊ $Cloud\_CVC3_{TRACK1/2}$
- ◊ ATC
- ◊ $KS_{UN}$

<u>Payload sent to the Issuer 180</u>

- ◊ 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**

**[0096]** 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)

**[0097]** 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 **$CVC3_{TRACK1/2}$** using:

◊ $Cloud\_CVC3_{TRACK1/2}$ (2 bytes)

◊ $UN_{READER}$ (4 bytes)

◊ ATC (2 bytes)

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

**Push Model Flows and Communication Sequences**

**[0098]** 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").

**[0099]** 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.

**[00100]** 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**

**Fn_GenCVC3**

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

**[0232]** A glossary of terms and acronyms described above and depicted in Figures 3-15 is provided in Table 2 below:

**Table 2**

| Symbol/acronym | Description |
|---|---|
| **2FA** | Two-factor Authentication |
| **AES** | Advanced Encryption Standard |
| **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 |
| **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 |
| **PUNATC** | Track *n* Bitmap for UN and ATC |

**Table 2**

| Symbol/acronym | Description |
|---|---|
| **PCVC3** | Track n Bitmap for CVC3 |
| **RC** | Return Code |
| **RFC** | Request for Comments |
| **SE** | Secure Element |
| **SHA** | Secure Hash Algorithm |
| **SIM** | Subscriber Identity Module |
| **SMS** | Short Message System |
| **UDID** | Unique Identifier |
| **UDOL** | Unpredictable Number Data Object List |
| **UN** | Unpredictable Number |
| $\oplus$ | XOR Operator |
| $\|\|$ | Concatenation Operator |

## IV. Exemplary Computer System Implementation

**[0233]** 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.

**[0234]** 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.

**[0235]** 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.

**[0236]** 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 1-15.

**[0237]** 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.

**[0238]** 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."

**[0239]** 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.

**[0240]** 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.

**[0241]** 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.

**[0242]** 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.

**[0243]** 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.

**[0244]** 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.

**[0245]** 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.

**[0246]** 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.

**[0247]** 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.).

**[0248]** 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

**[0249]** 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.

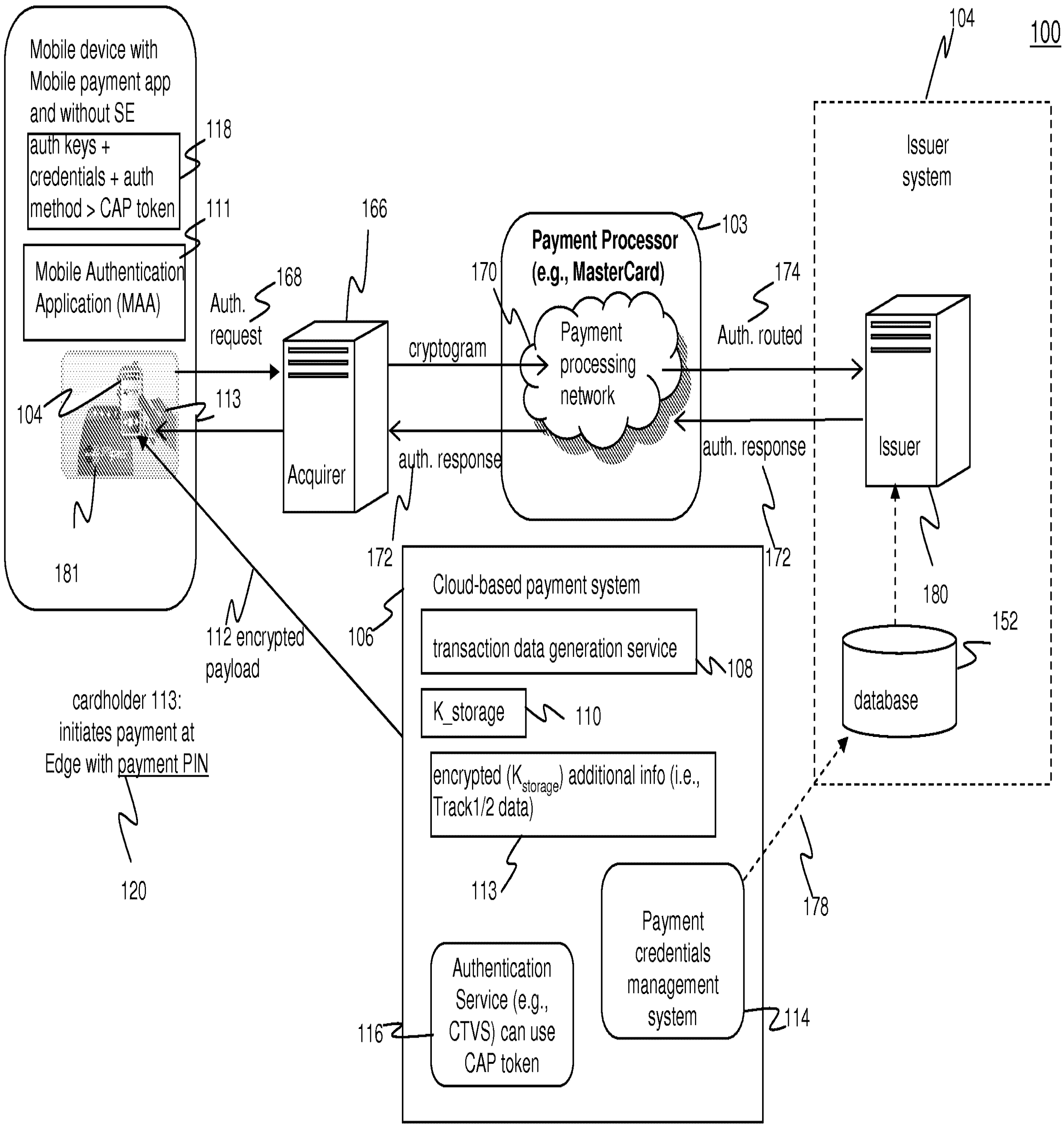
27. The method of claim 23, wherein the time controls comprise day of week controls for the tokens.

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.



100
104
Issuer system
Mobile device with Mobile payment app and without SE
auth keys + credentials + auth method > CAP token
118
111
Mobile Authentication Application (MAA)
104
113
181
Auth. request
168
166
Acquirer
cryptogram
auth. response
172
170
Payment Processor (e.g., MasterCard)
103
Payment processing network
174
Auth. routed
auth. response
172
Issuer
180
152
database
112 encrypted payload
cardholder 113: initiates payment at Edge with payment PIN
120
Cloud-based payment system
106
transaction data generation service
108
K_storage
110
encrypted (K_storage) additional info (i.e., Track1/2 data)
113
Payment credentials management system
178
114
Authentication Service (e.g., CTVS) can use CAP token
116


FIG. 1

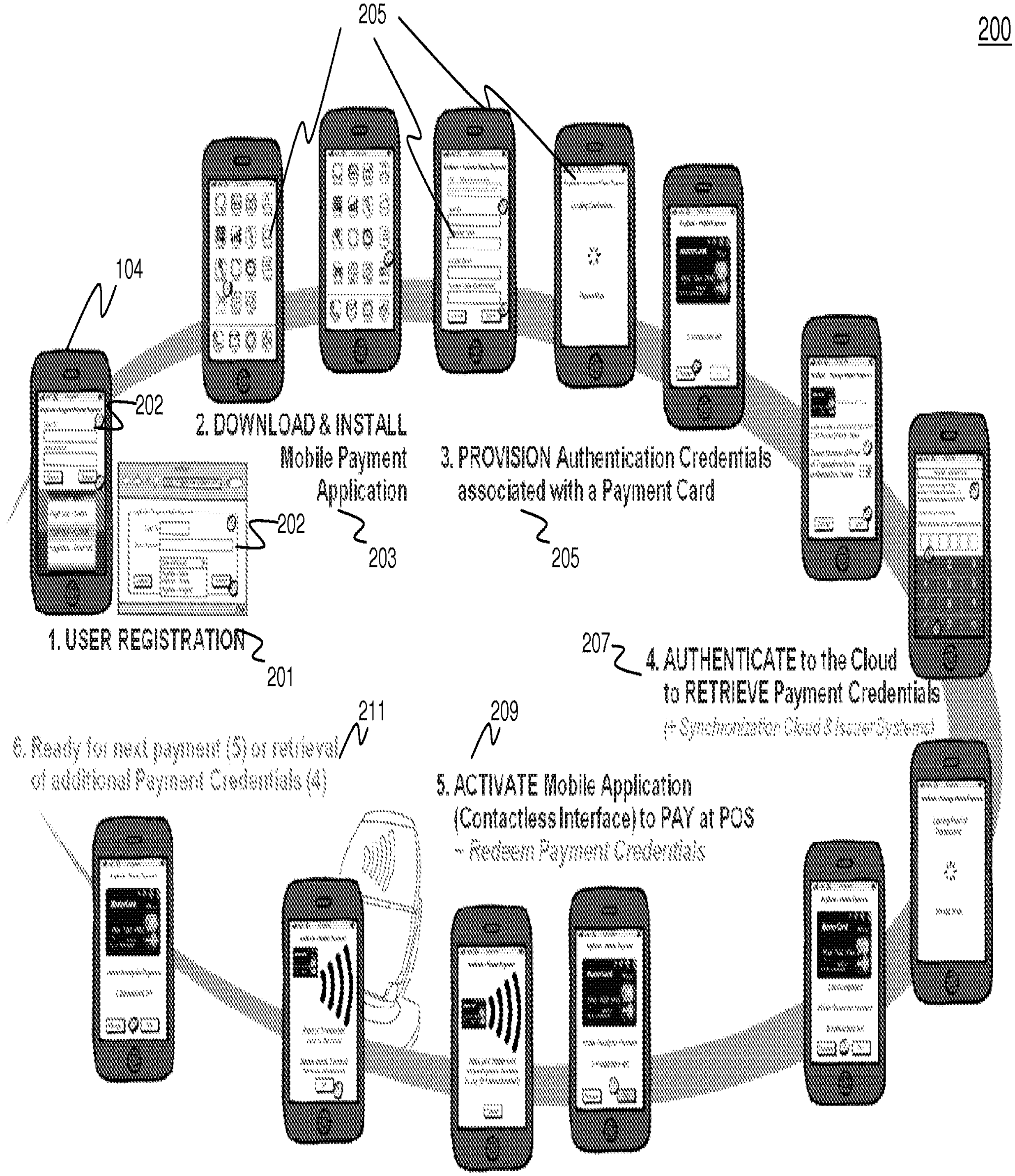
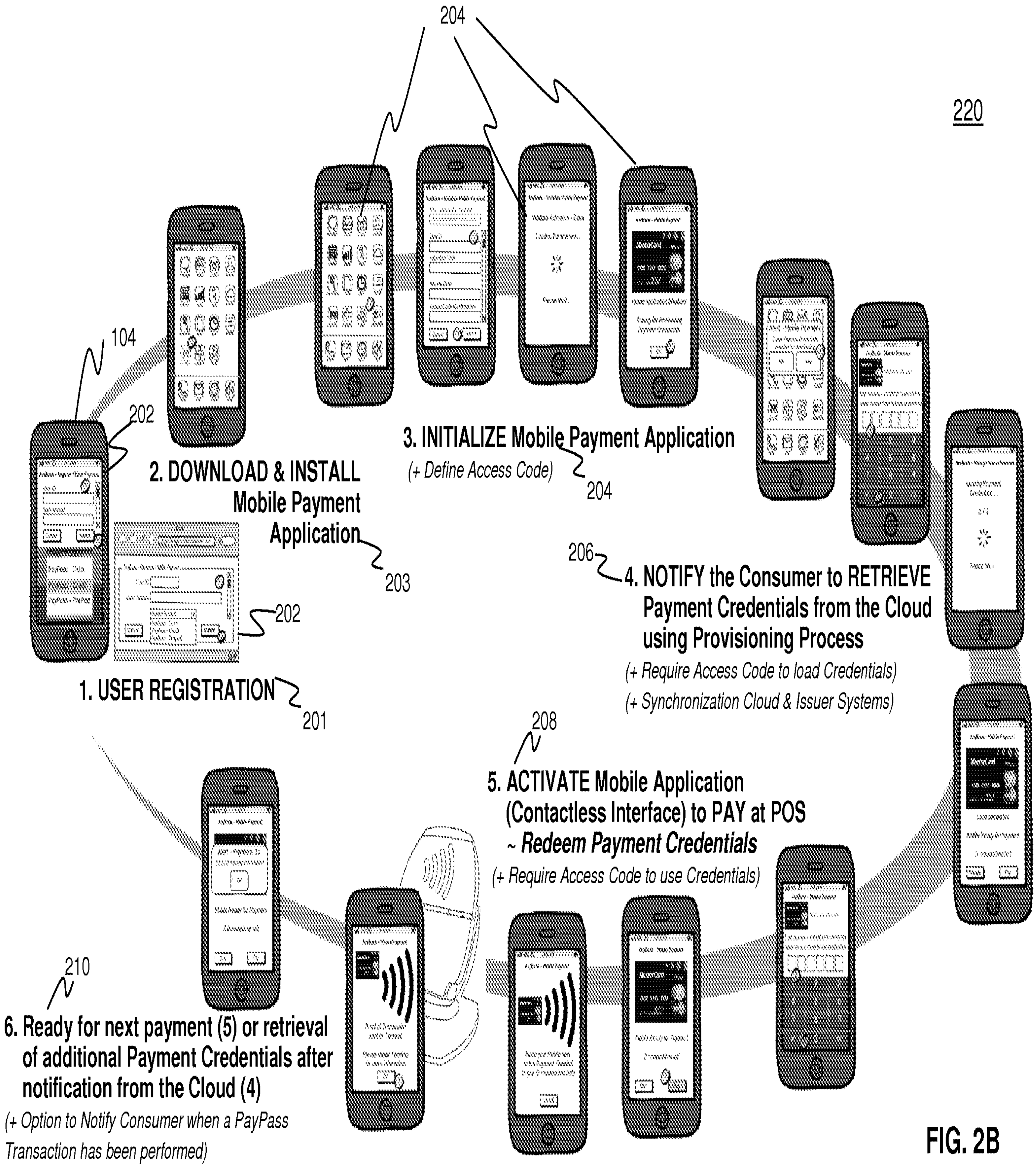
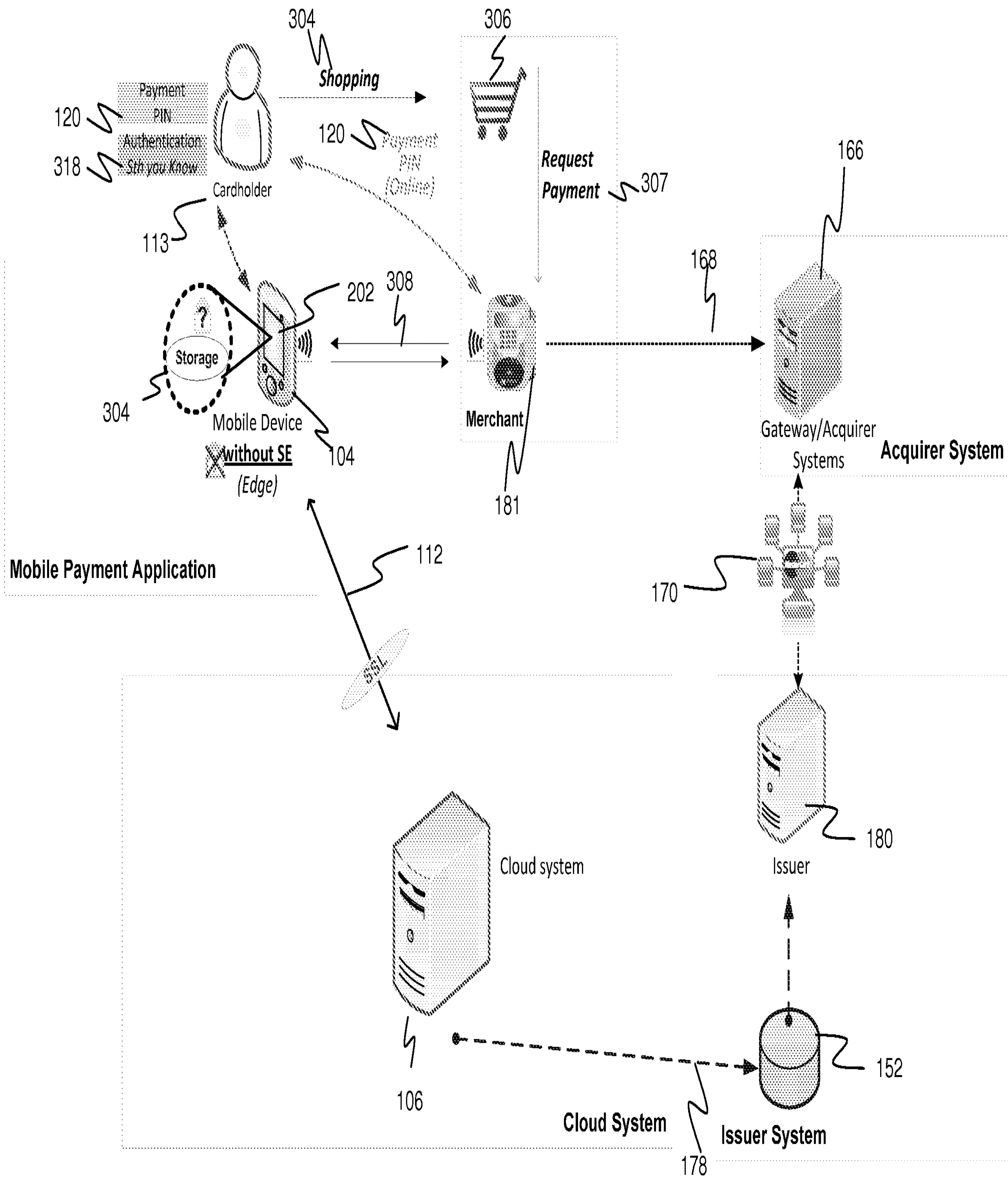

205
200
104
202
2. DOWNLOAD & INSTALL Mobile Payment Application
3. PROVISION Authentication Credentials associated with a Payment Card
202
203
205
1. USER REGISTRATION
201
207
4. AUTHENTICATE to the Cloud to RETRIEVE Payment Credentials
(+ Synchronization Cloud & Issuer Systems)
211
209
6. Ready for 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
202
1. USER REGISTRATION
201
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)
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
Authentication
Sth you Know
120
318
120
Payment
PIN
(Online)
Cardholder
Request
Payment
307
166
113
168
202
308
?
Storage
304
Mobile Device
without SE
(Edge)
104
Merchant
181
Gateway/Acquirer
Systems
Acquirer System
112
Mobile Payment Application
170
SSL
180
Issuer
Cloud system
106
152
Cloud System
Issuer System
178


FIG. 3

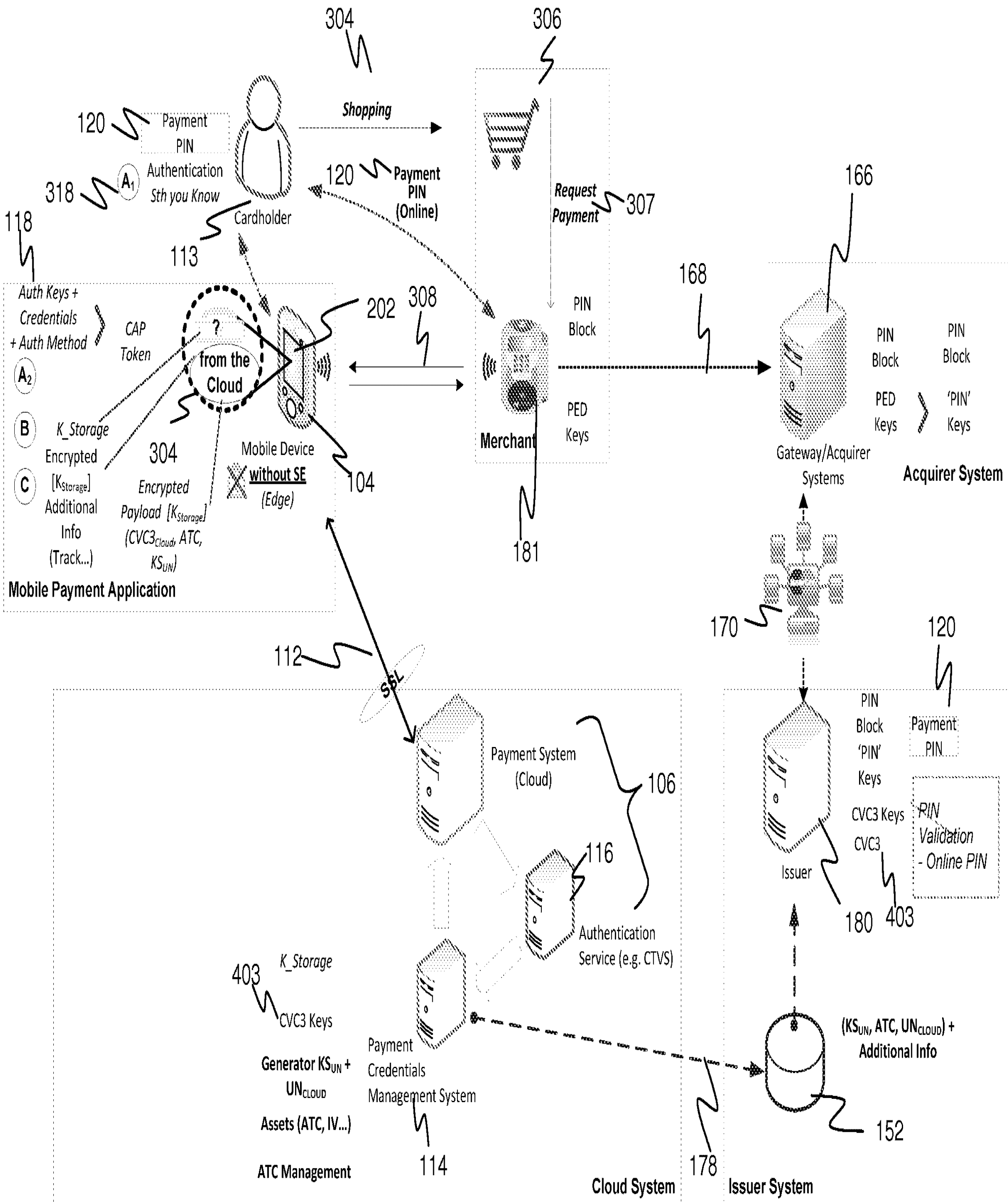
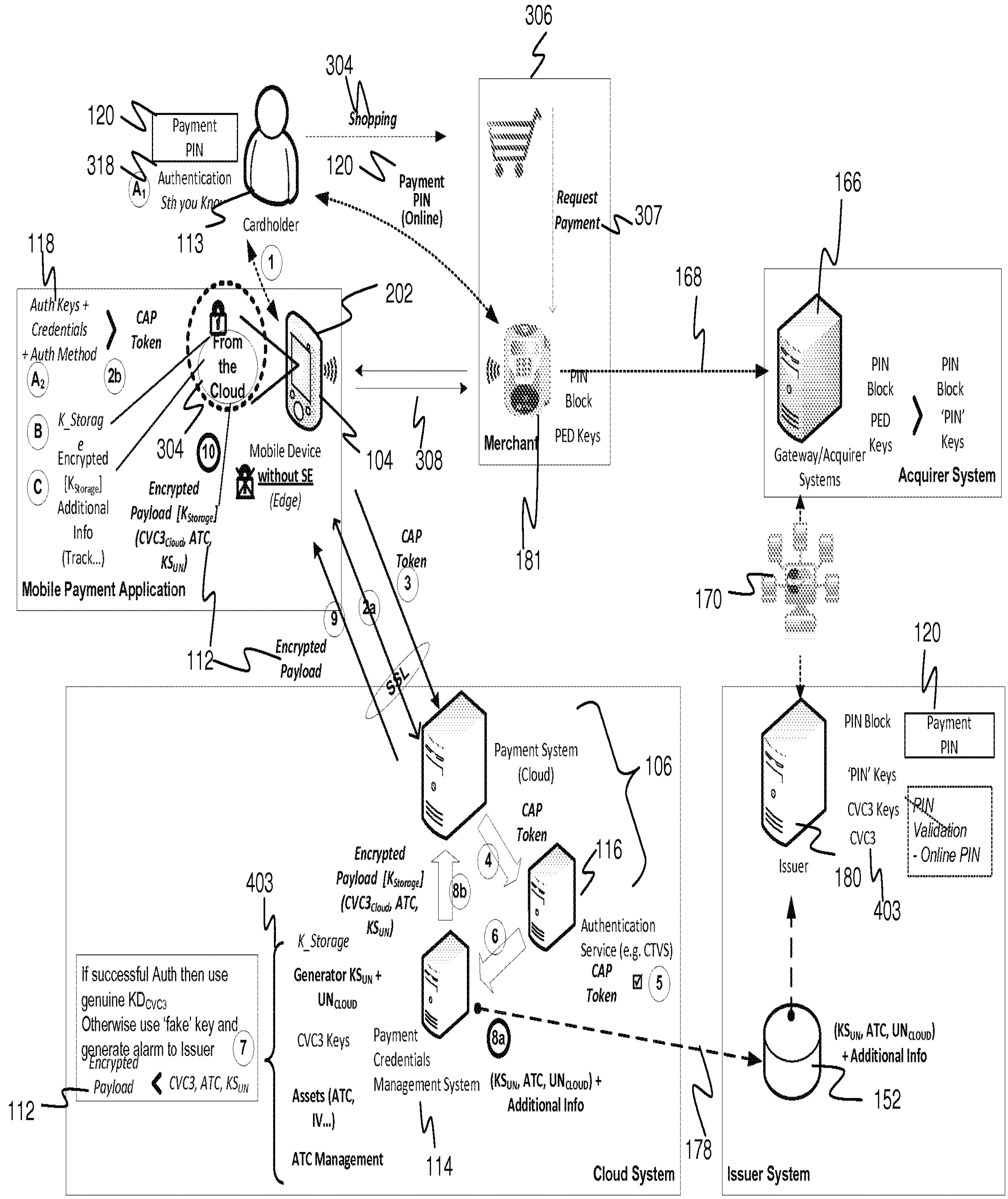

304
306
Shopping
120
Payment
PIN
318
Authentication
Sth you Know
Cardholder
120
Payment
PIN
(Online)
Request
Payment
307
166
118
113
Auth Keys +
Credentials
+ Auth Method
CAP
Token
?
from the
Cloud
202
308
168
PIN
Block
PIN
Block
PIN
Block
PED
Keys
'PIN'
Keys
PED
Keys
Merchant
K_Storage
Encrypted
[K_Storage]
Additional
Info
(Track...)
304
Encrypted
Payload [K_Storage]
(CVC3_Cloud, ATC,
KS_UN)
Mobile Device
without SE
(Edge)
104
181
Gateway/Acquirer
Systems
Acquirer System
Mobile Payment Application
170
120
112
SSL
PIN
Block
'PIN'
Keys
Payment
PIN
Payment System
(Cloud)
106
CVC3 Keys
CVC3
PIN
Validation
- Online PIN
116
Issuer
180
403
Authentication
Service (e.g. CTVS)
403
K_Storage
CVC3 Keys
Generator KS_UN +
UN_CLOUD
Assets (ATC, IV...)
ATC Management
Payment
Credentials
Management System
114
(KS_UN, ATC, UN_CLOUD) +
Additional Info
152
178
Cloud System
Issuer System

FIG. 4



306
304
120
Payment
PIN
318
Authentication
Sth you Know
Shopping
120
Payment
PIN
(Online)
Cardholder
113
Request
Payment
307
166
118
Auth Keys +
Credentials
+ Auth Method
CAP
Token
2b
From
the
Cloud
202
1
168
K_Storage
Encrypted
[K_Storage]
Additional
Info
(Track...)
304
10
Encrypted
Payload [K_Storage]
(CVC3_Cloud, ATC,
KS_UN)
Mobile Device
without SE
(Edge)
104
308
PIN
Block
PED Keys
Merchant
PIN
Block
PED
Keys
PIN
Block
'PIN'
Keys
Gateway/Acquirer
Systems
Acquirer System
Mobile Payment Application
CAP
Token
3
2a
9
181
170
112
Encrypted
Payload
SSL
120
Payment System
(Cloud)
106
CAP
Token
116
PIN Block
Payment
PIN
'PIN' Keys
CVC3 Keys
CVC3
PIN
Validation
- Online PIN
Issuer
180
403
403
Encrypted
Payload [K_Storage]
(CVC3_Cloud, ATC,
KS_UN)
8b
4
6
K_Storage
Generator KS_UN +
UN_CLOUD
Authentication
Service (e.g. CTVS)
CAP
Token
5
If successful Auth then use
genuine KD_CVC3
Otherwise use 'fake' key and
generate alarm to Issuer
7
Encrypted
Payload
CVC3, ATC, KS_UN
CVC3 Keys
Payment
Credentials
Management System
8a
(KS_UN, ATC, UN_CLOUD)
+ Additional Info
112
Assets (ATC,
IV...)
(KS_UN, ATC, UN_CLOUD) +
Additional Info
152
ATC Management
114
178
Cloud System
Issuer System


FIG. 5

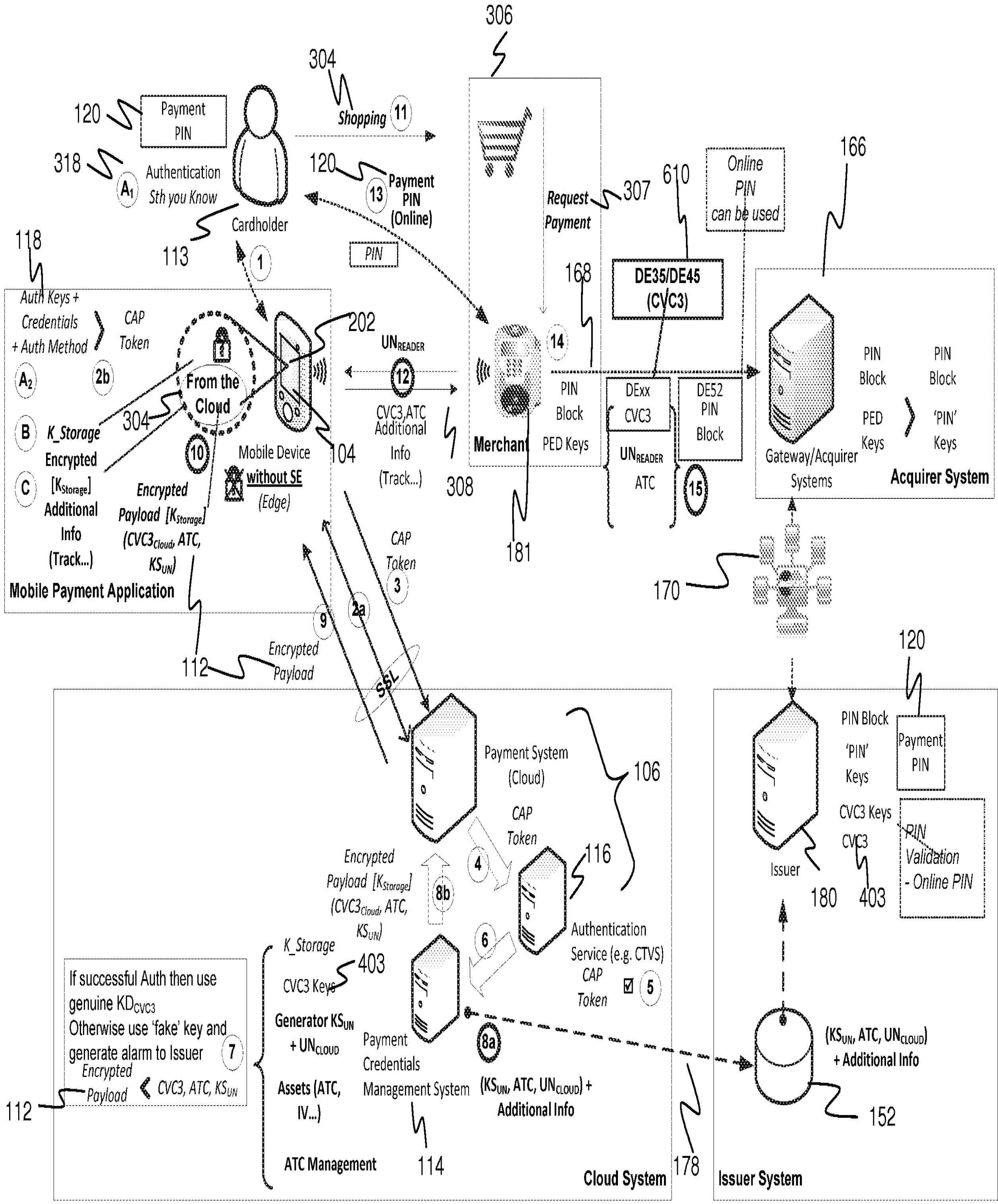
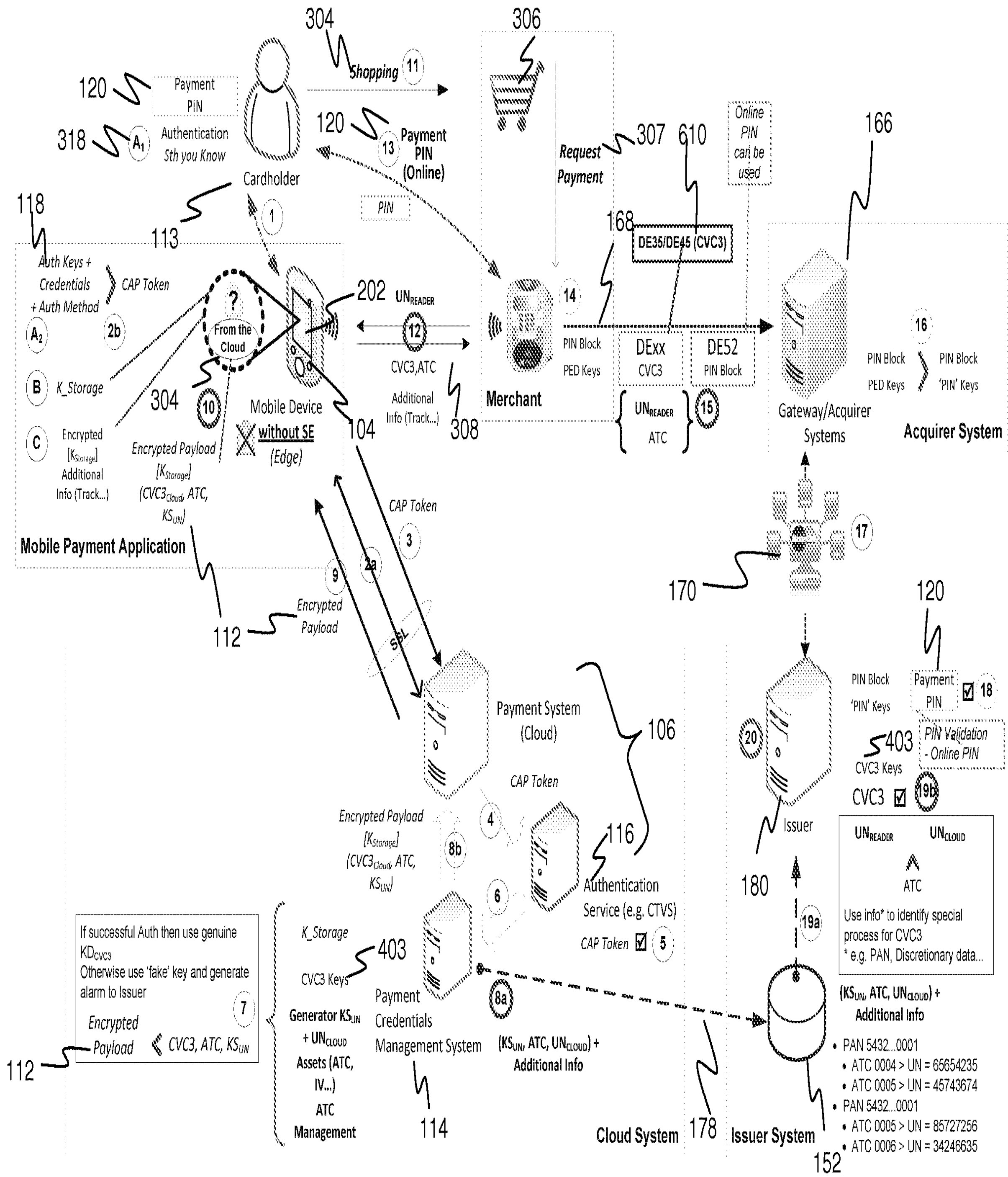
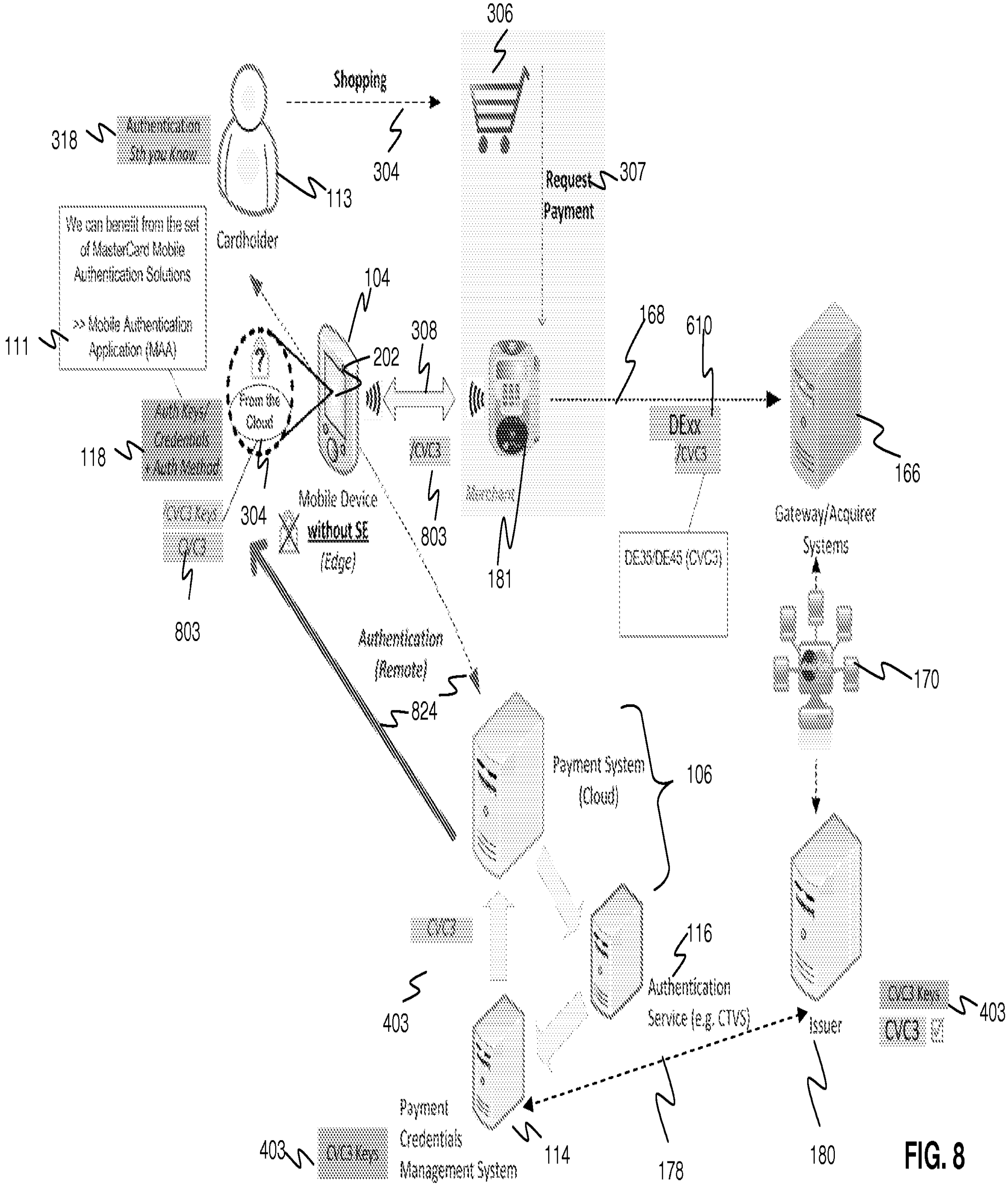

306
304
Shopping
11
120
Payment PIN
318
A1
Authentication
Sth you Know
Cardholder
113
120
13
Payment PIN (Online)
PIN
Request Payment
307
610
Online PIN can be used
166
118
Auth Keys + Credentials + Auth Method
CAP Token
A2
2b
From the Cloud
202
UNREADER
12
CVC3,ATC Additional Info (Track...)
308
168
DE35/DE45 (CVC3)
14
Merchant
PIN Block
PED Keys
DExx CVC3
DE52 PIN Block
UNREADER ATC
15
PIN Block
PED Keys
'PIN' Keys
Gateway/Acquirer Systems
Acquirer System
B
K_Storage Encrypted
304
C
[KStorage] Additional Info (Track...)
10
Encrypted Payload [KStorage] (CVC3Cloud, ATC, KSUN)
Mobile Device without SE (Edge)
104
Mobile Payment Application
CAP Token
3
2a
9
181
170
112
Encrypted Payload
SSL
120
Payment System (Cloud)
106
CAP Token
4
116
PIN Block
'PIN' Keys
Payment PIN
CVC3 Keys
CVC3
PIN Validation - Online PIN
Issuer
180
403
Encrypted Payload [KStorage] (CVC3Cloud, ATC, KSUN)
8b
6
Authentication Service (e.g. CTVS)
CAP Token
5
K_Storage
403
CVC3 Keys
Generator KSUN + UNCLOUD
Assets (ATC, IV...)
ATC Management
If successful Auth then use genuine KDCVC3
Otherwise use 'fake' key and generate alarm to Issuer
7
Encrypted Payload
CVC3, ATC, KSUN
112
Payment Credentials Management System
8a
(KSUN, ATC, UNCLOUD) + Additional Info
114
(KSUN, ATC, UNCLOUD) + Additional Info
152
Cloud System
178
Issuer System

FIG. 6



304
Shopping
11
306
120
Payment PIN
318
A1
Authentication
Sth you Know
Cardholder
120
Payment PIN (Online)
13
Request Payment
307
610
Online PIN can be used
166
PIN
118
113
1
168
DE35/DE45 (CVC3)
Auth Keys + Credentials + Auth Method
CAP Token
2b
A2
?
From the Cloud
202
UNREADER
12
14
PIN Block
PED Keys
DExx
CVC3
DE52
PIN Block
16
PIN Block
PED Keys
PIN Block
'PIN' Keys
B
K_Storage
304
10
Mobile Device
without SE
(Edge)
CVC3,ATC
Additional Info (Track...)
308
Merchant
UNREADER
ATC
15
Gateway/Acquirer Systems
Acquirer System
C
Encrypted [KStorage] Additional Info (Track...)
Encrypted Payload [KStorage] (CVC3Cloud, ATC, KSUN)
104
CAP Token
3
Mobile Payment Application
2a
9
17
170
Encrypted Payload
112
SSL
120
Payment System (Cloud)
106
PIN Block
'PIN' Keys
Payment PIN
18
20
403
PIN Validation - Online PIN
CVC3 Keys
CVC3
19b
CAP Token
Encrypted Payload [KStorage] (CVC3Cloud, ATC, KSUN)
4
8b
116
Issuer
UNREADER
UNCLOUD
ATC
180
6
Authentication Service (e.g. CTVS)
If successful Auth then use genuine KDCVC3
Otherwise use 'fake' key and generate alarm to Issuer
CAP Token
5
19a
Use info* to identify special process for CVC3
* e.g. PAN, Discretionary data...
K_Storage
403
CVC3 Keys
7
Encrypted Payload
CVC3, ATC, KSUN
112
Generator KSUN + UNCLOUD
Assets (ATC, IV...)
ATC Management
Payment Credentials Management System
8a
(KSUN, ATC, UNCLOUD) + Additional Info
(KSUN, ATC, UNCLOUD) + Additional Info
PAN 5432...0001
ATC 0004 > UN = 65654235
ATC 0005 > UN = 45743674
PAN 5432...0001
ATC 0005 > UN = 85727256
ATC 0006 > UN = 34246635
114
Cloud System
178
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
308
202
168
610
From the Cloud
Auth Keys/ Credentials + Auth Method
118
CVC3
DExx
/CVC3
166
304
Mobile Device
without SE
(Edge)
803
Merchant
CVC3 Keys
CVC3
803
181
DE35/DE45 (CVC3)
Gateway/Acquirer Systems
Authentication
(Remote)
824
170
Payment System
(Cloud)
106
CVC3
403
116
Authentication Service (e.g. CTVS)
Issuer
CVC3 Keys
CVC3
403
Payment Credentials Management System
403
CVC3 Keys
114
178
180

FIG. 8

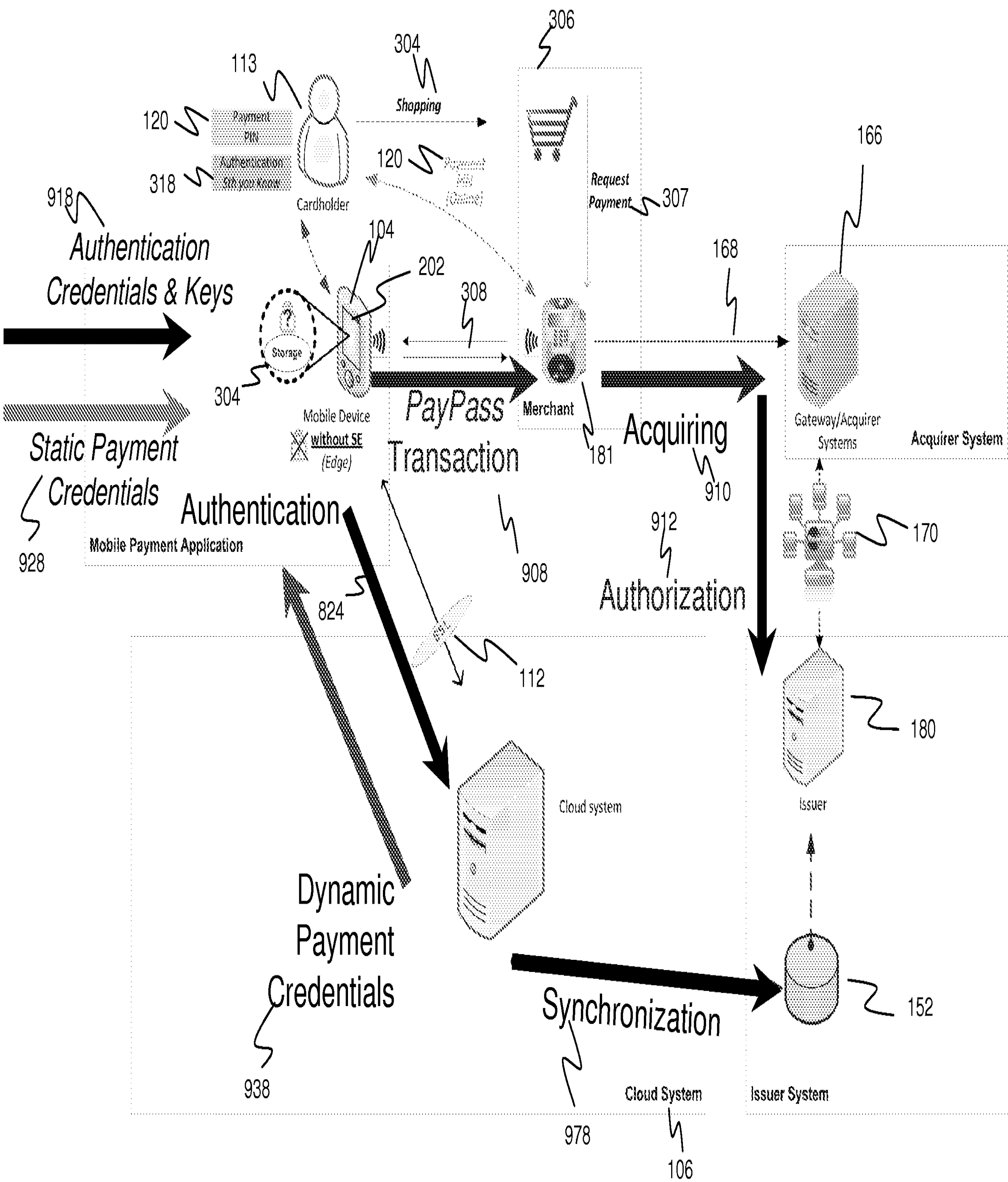
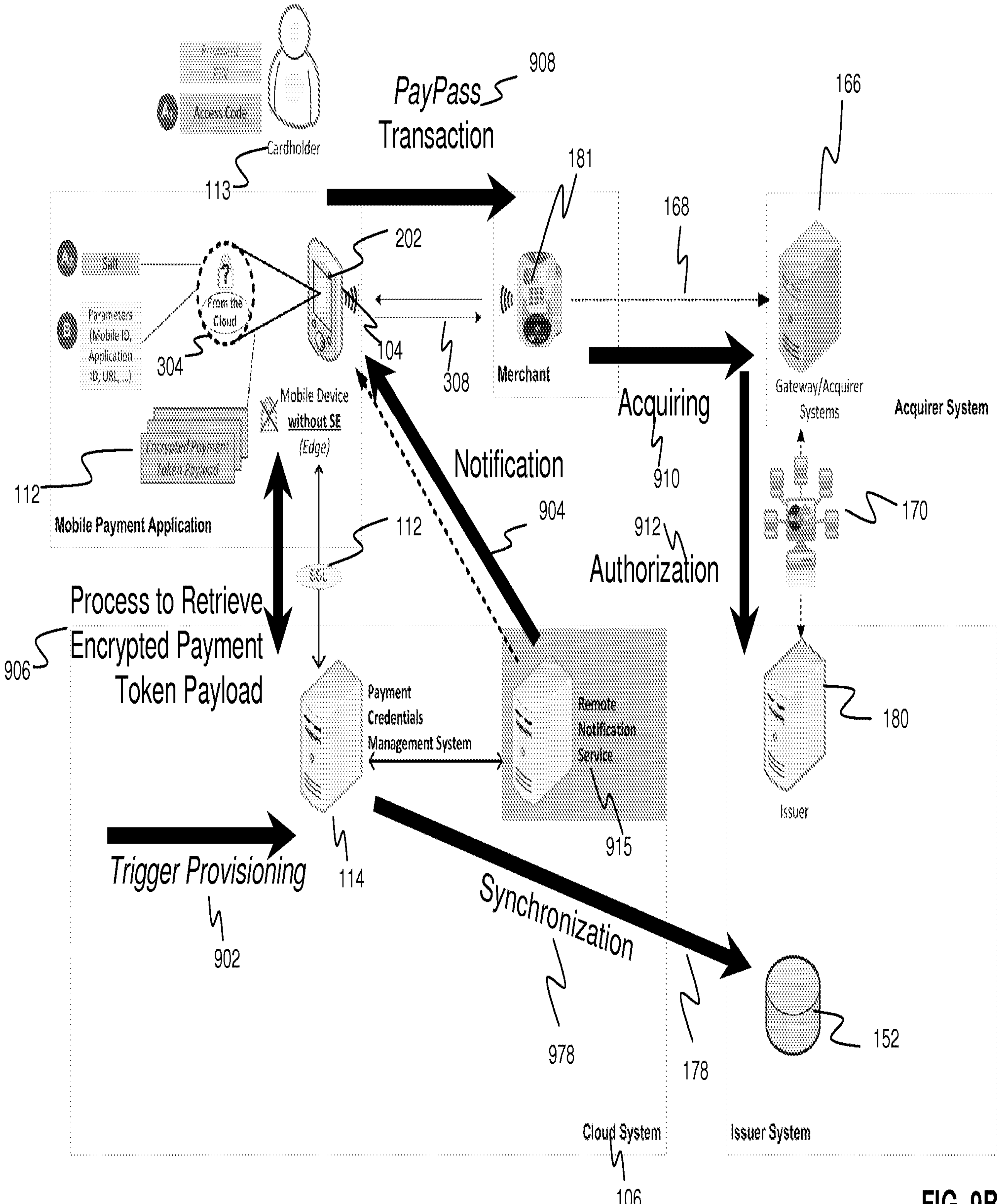
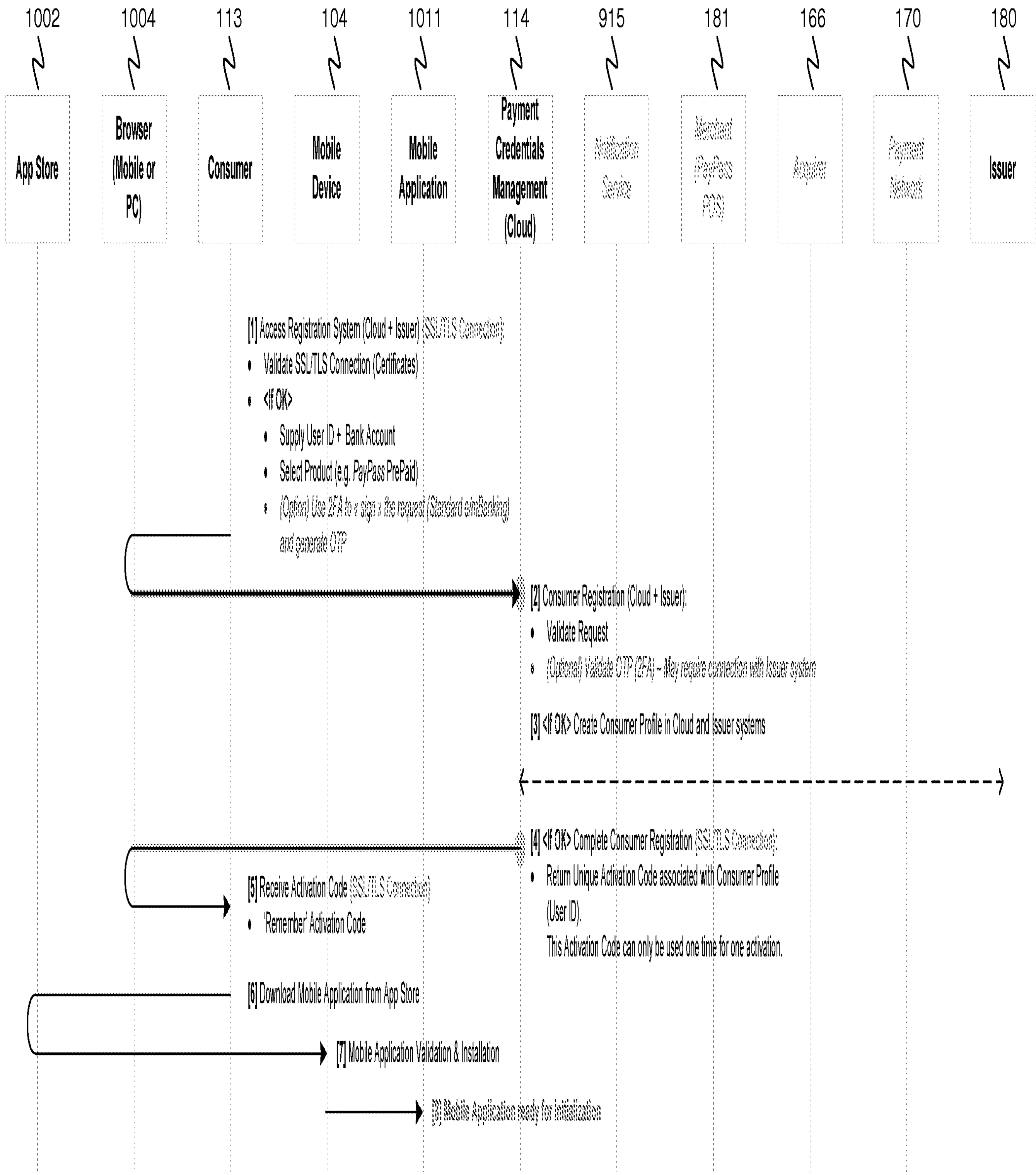

306
304
Shopping
113
120
Payment PIN
318
Authentication Sth you Know
Cardholder
120
Request Payment
307
166
918
Authentication Credentials & Keys
104
202
168
308
Storage
304
Mobile Device
without SE
(Edge)
PayPass Transaction
Merchant
181
Acquiring
910
Gateway/Acquirer Systems
Acquirer System
Static Payment Credentials
928
Mobile Payment Application
Authentication
824
908
912
Authorization
170
112
180
Issuer
Cloud system
Dynamic Payment Credentials
938
Synchronization
978
152
Cloud System
106
Issuer System

FIG. 9A



Access Code
Cardholder
PayPass
Transaction
908
166
113
181
168
Salt
From the Cloud
202
Parameters (Mobile ID, Application ID, URL, ...)
304
104
308
Merchant
Acquiring
Gateway/Acquirer Systems
Acquirer System
Mobile Device without SE (Edge)
Notification
910
112
Mobile Payment Application
112
904
912
170
Authorization
Process to Retrieve Encrypted Payment Token Payload
906
SSL
Payment Credentials Management System
Remote Notification Service
180
Issuer
915
Trigger Provisioning
114
Synchronization
902
152
978
178
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)
Notification Service
Merchant (PayPass POS)
Acquirer
Payment Network
Issuer
[1] Access Registration System (Cloud + Issuer) (SSL/TLS Connection)
• Validate SSL/TLS Connection (Certificates)
• <If OK>
• Supply User ID + Bank Account
• Select Product (e.g. PayPass PrePaid)
• (Option) Use 2FA to « sign » the request (Standard e/mBanking) and generate OTP
[2] Consumer Registration (Cloud + Issuer):
• Validate Request
• (Optional) Validate OTP (2FA) ~ May require connection with Issuer system
[3] <If OK> Create Consumer Profile in Cloud and Issuer systems
[4] <If OK> Complete Consumer Registration (SSL/TLS Connection)
• Return Unique Activation Code associated with Consumer Profile (User ID).
This Activation Code can only be used one time for one activation.
[5] Receive Activation Code (SSL/TLS Connection)
• 'Remember' Activation Code
[6] Download Mobile Application from App Store
[7] Mobile Application Validation & Installation
[8] Mobile Application ready for Initialization


**FIG. 10**

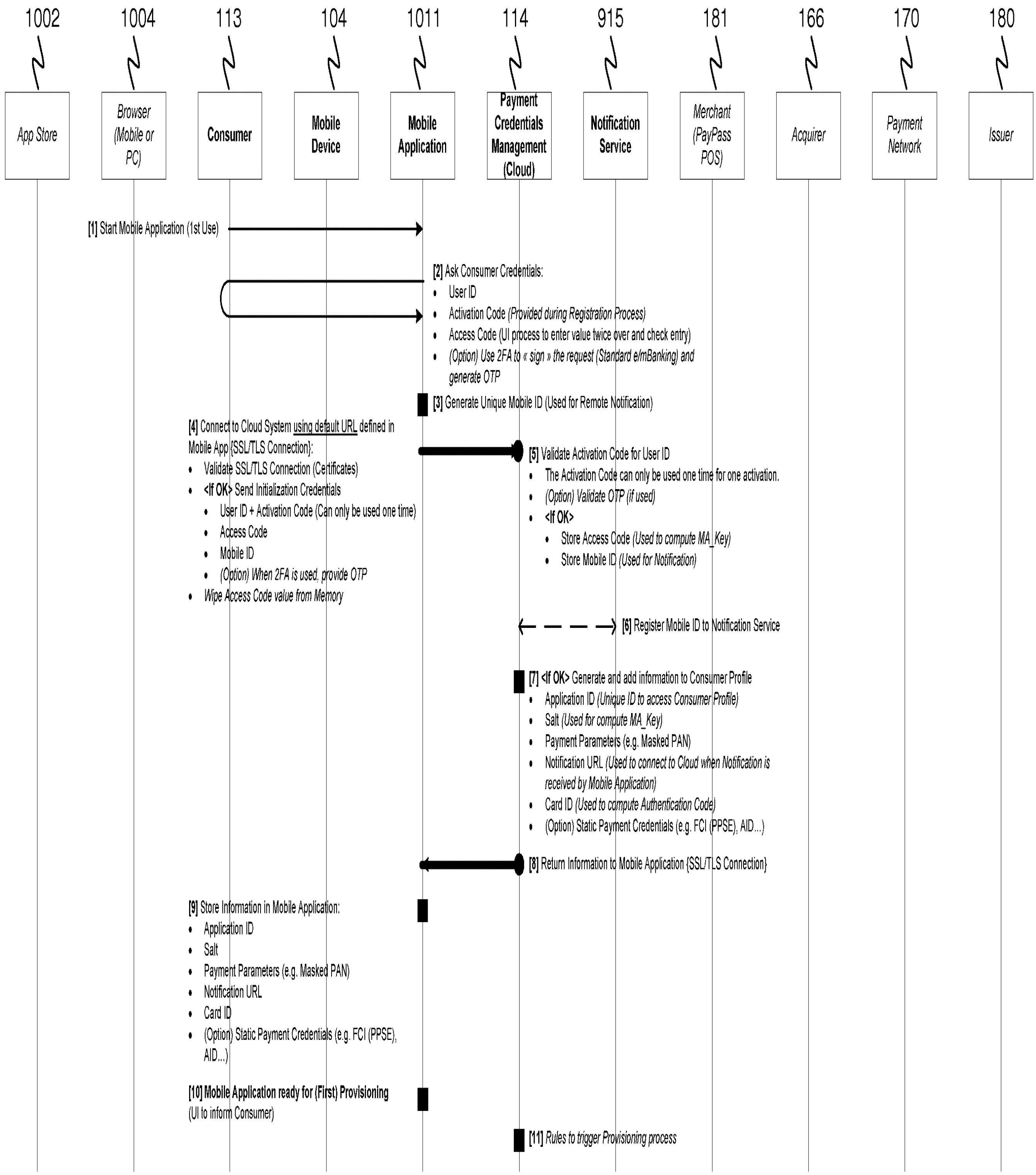
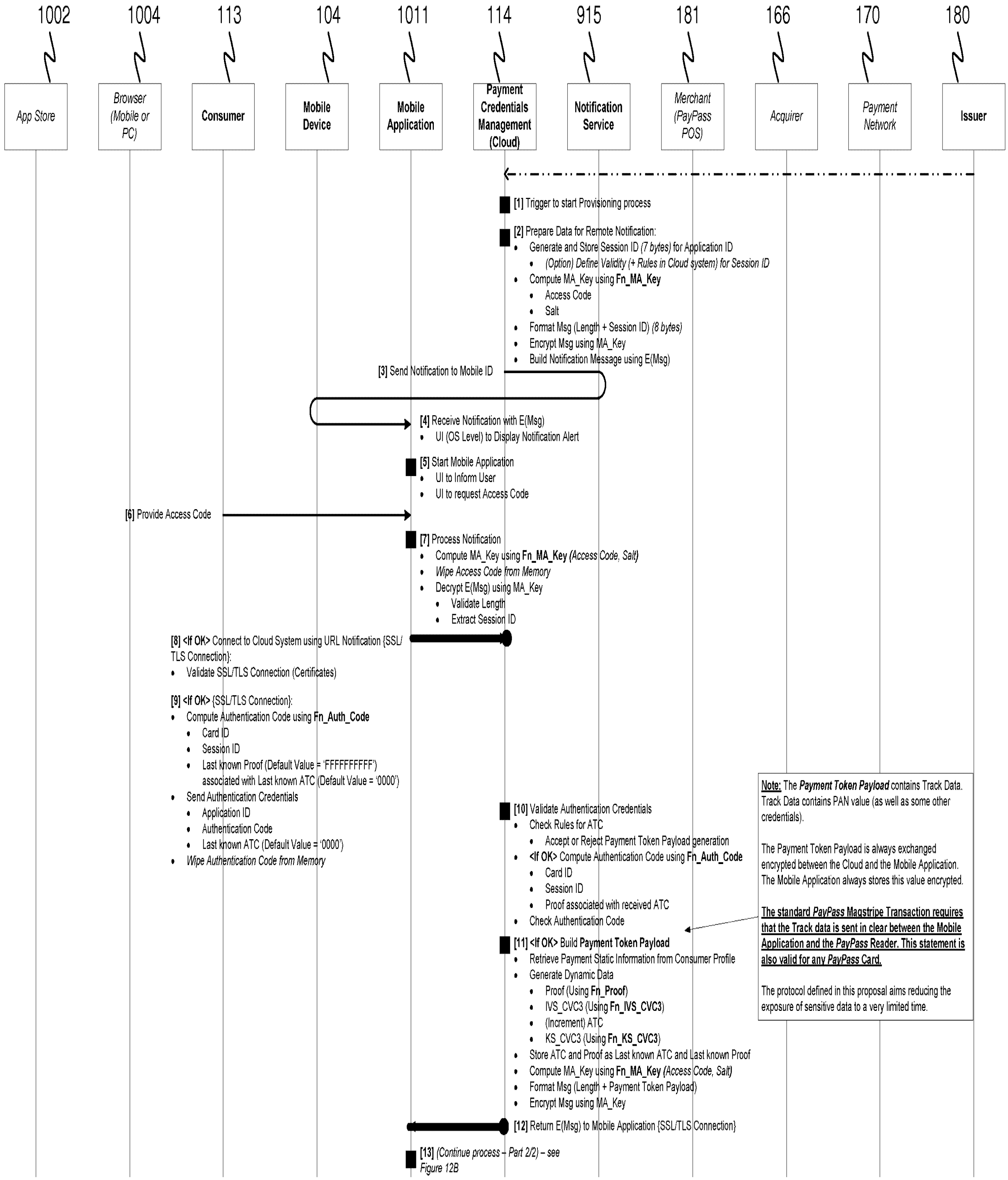
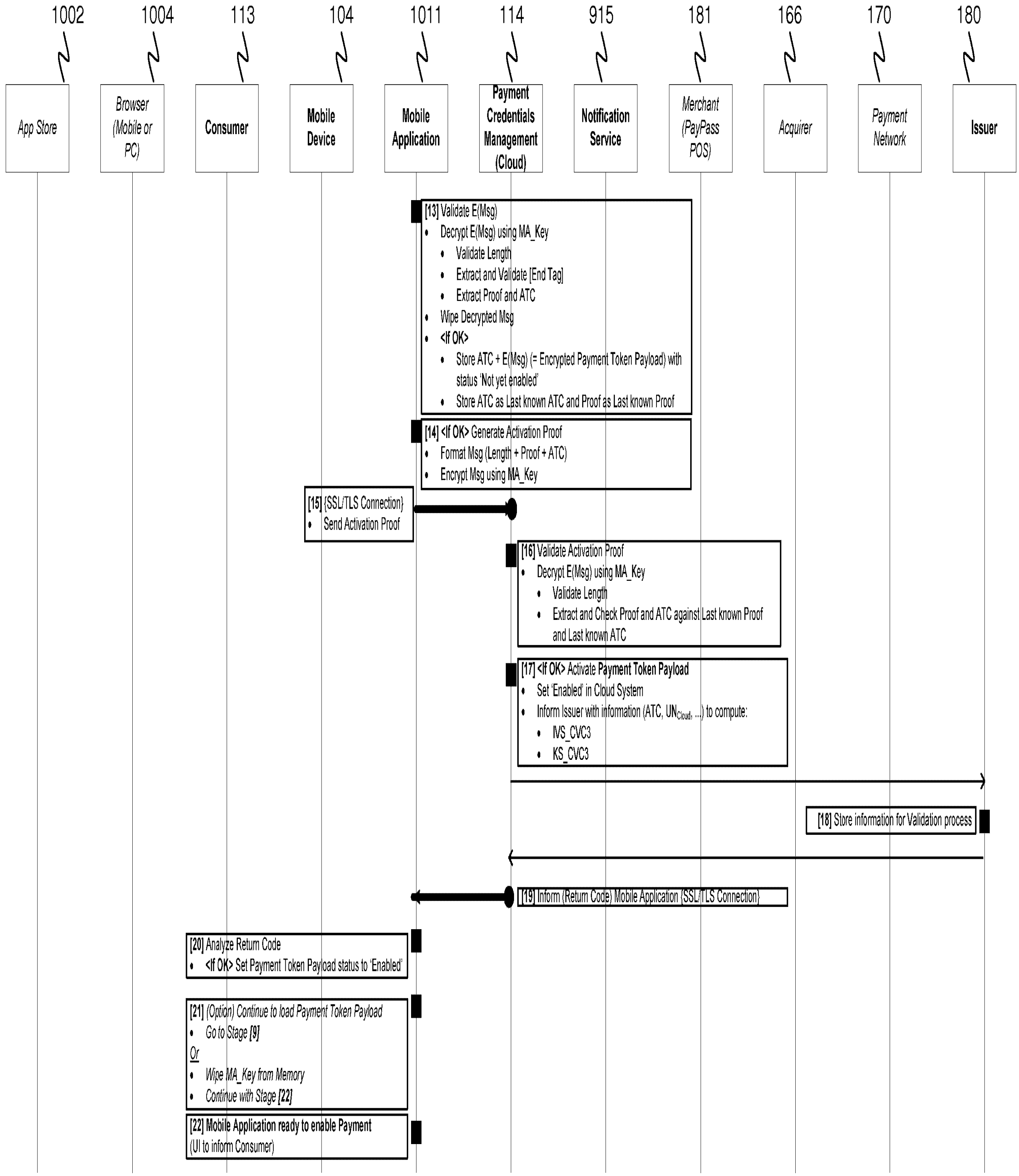

1002
1004
113
104
1011
114
915
181
166
170
180
App Store
Browser (Mobile or PC)
Consumer
Mobile Device
Mobile Application
Payment Credentials Management (Cloud)
Notification Service
Merchant (PayPass POS)
Acquirer
Payment Network
Issuer
[1] Start Mobile Application (1st Use)
[2] Ask Consumer Credentials:
• 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 = 'FFFFFFFFFF') 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.

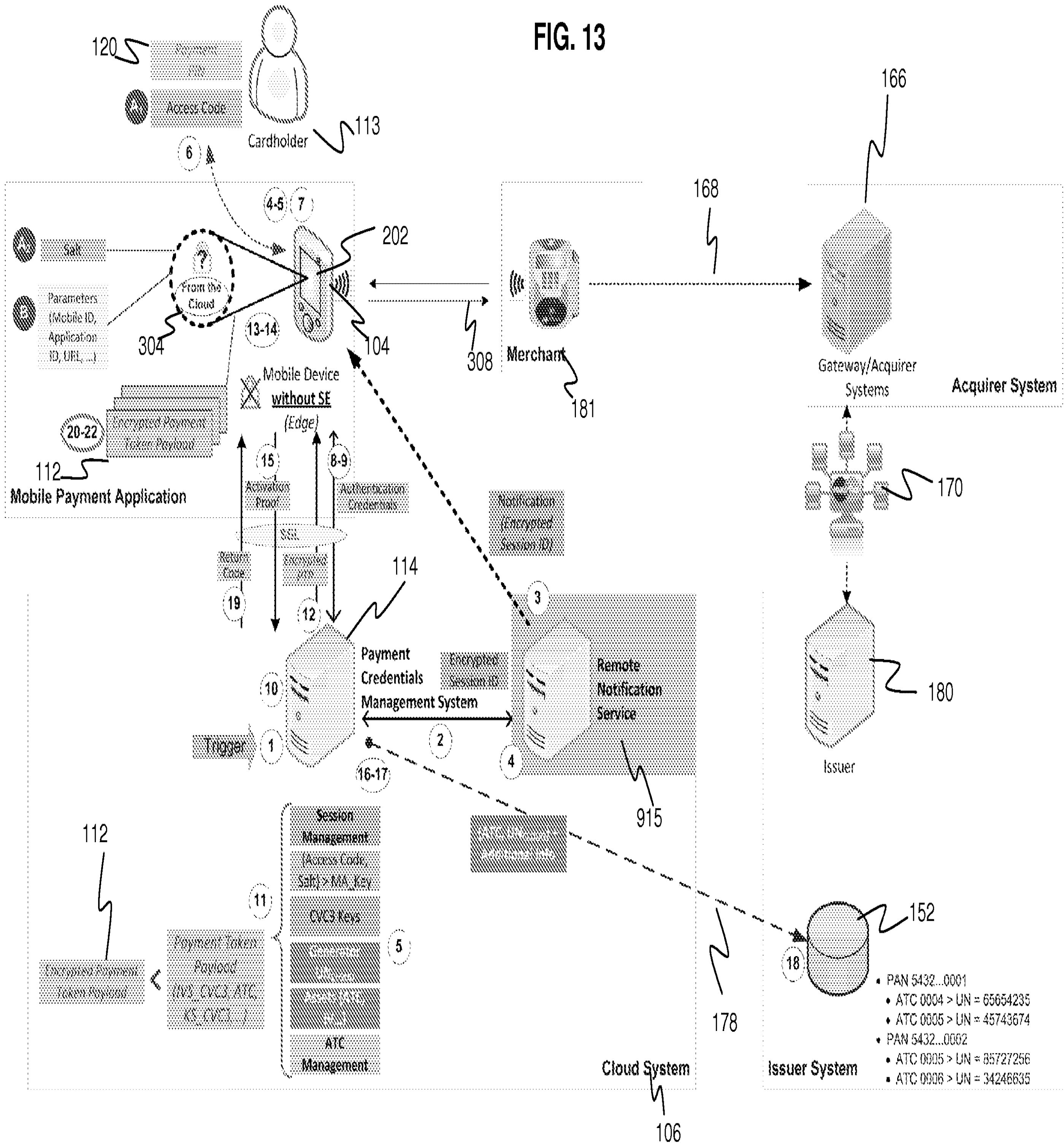

FIG. 12A



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
Payment
PIN
Access Code
Cardholder
113
166
6
4-5
7
202
168
Salt
?
From the Cloud
Parameters (Mobile ID, Application ID, URL, ...)
304
13-14
104
308
Merchant
181
Gateway/Acquirer Systems
Acquirer System
Mobile Device
without SE
(Edge)
20-22
Encrypted Payment Token Payload
112
Mobile Payment Application
15
Activation Proof
8-9
Authentication Credentials
170
Notification (Encrypted Session ID)
SSL
Return Code
Encrypted PIN
114
19
12
3
Payment Credentials Management System
Encrypted Session ID
Remote Notification Service
180
10
Trigger
1
2
4
16-17
Issuer
915
112
Session Management
(Access Code, Salt) > MA_Key
CVC3 Keys
11
5
Payment Token Payload (IVS_CVC3, ATC, KS_CVC3, ...)
Encrypted Payment Token Payload
ATC Management
152
18
178
PAN 5432...0001
ATC 0004 > UN = 65654235
ATC 0005 > UN = 45743674
PAN 5432...0002
ATC 0005 > UN = 85727256
ATC 0006 > UN = 34246635
Cloud System
Issuer System
106

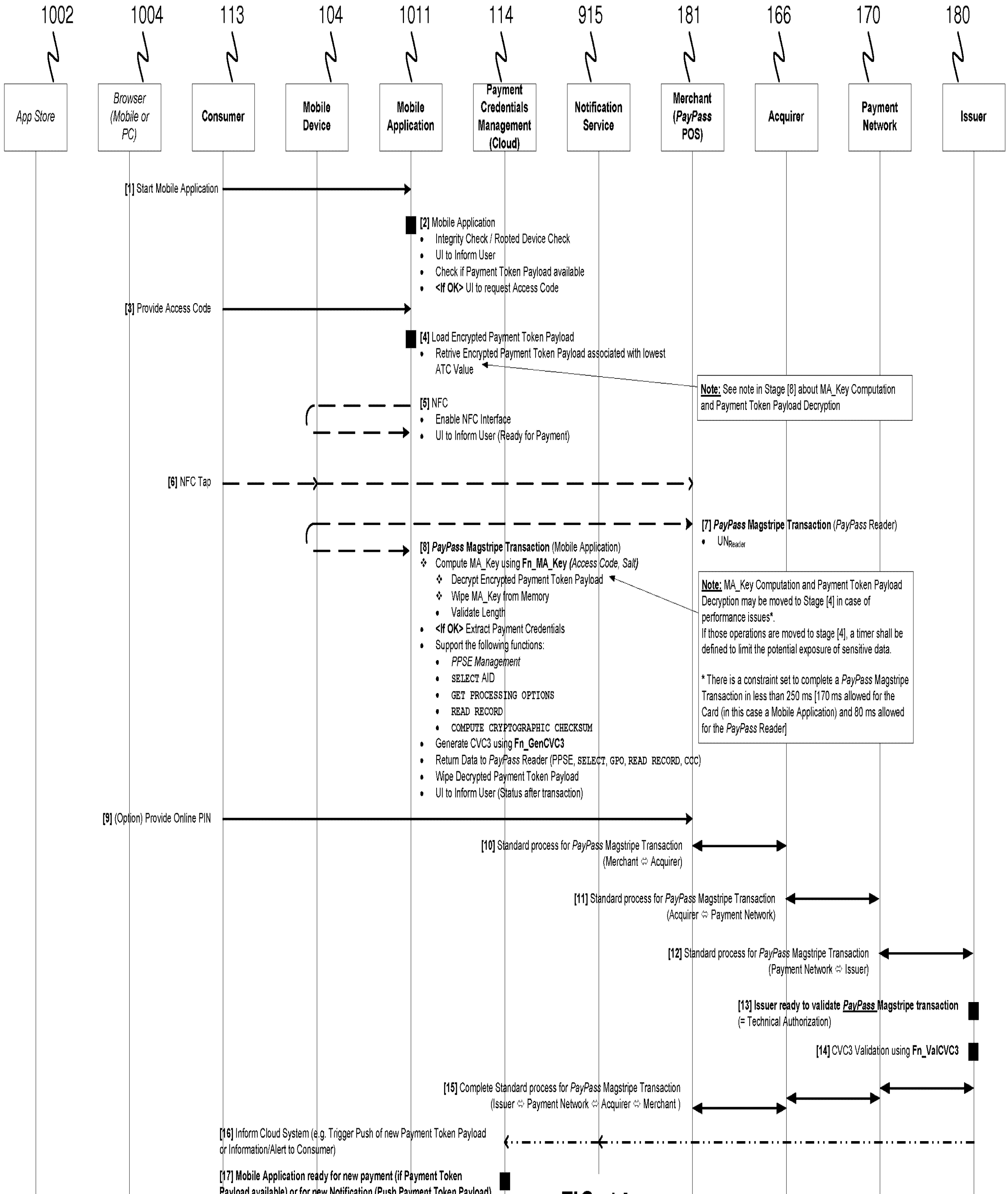

1002
1004
113
104
1011
114
915
181
166
170
180
App Store
Browser (Mobile or PC)
Consumer
Mobile Device
Mobile Application
Payment Credentials Management (Cloud)
Notification Service
Merchant (PayPass POS)
Acquirer
Payment Network
Issuer
[1] Start Mobile Application
[2] Mobile Application
Integrity Check / 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
Retrive 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)
UNReader
[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

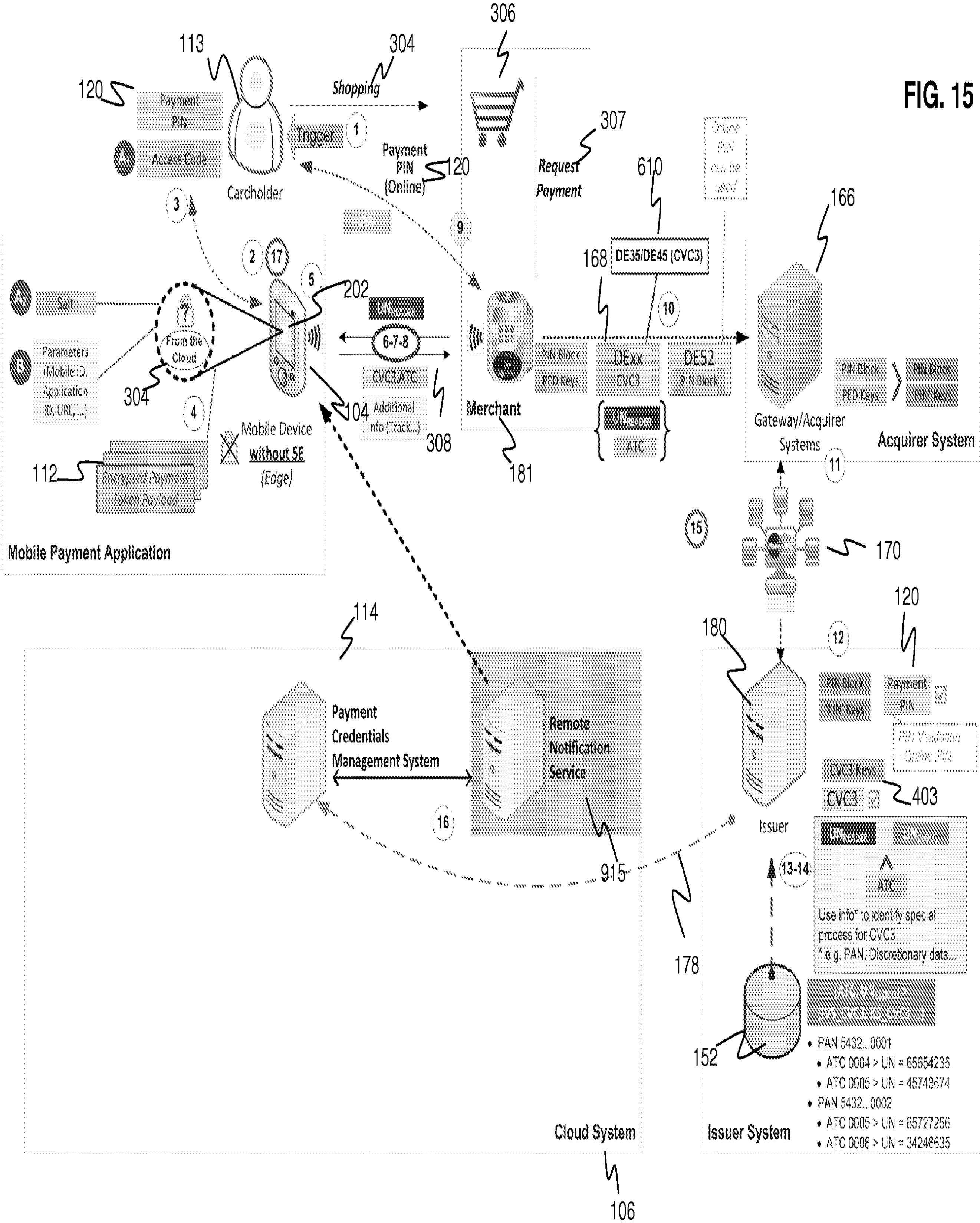

306
113
304
120
Payment PIN
Access Code
Shopping
Trigger
1
Cardholder
Payment PIN (Online)
Request Payment
307
610
166
3
9
DE35/DE45 (CVC3)
168
2
17
5
202
Salt
From the Cloud
6-7-8
10
Parameters (Mobile ID, Application ID, URL, ...)
304
CVC3,ATC
PIN Block
PED Keys
DExx CVC3
DE52 PIN Block
PIN Block
PED Keys
104
Additional Info (Track...)
308
Merchant
ATC
Gateway/Acquirer Systems
Acquirer System
4
Mobile Device without SE (Edge)
112
Encrypted Payment Token Payloads
181
11
15
170
Mobile Payment Application
114
120
180
12
Payment PIN
Payment Credentials Management System
Remote Notification Service
CVC3 Keys
CVC3
403
16
Issuer
915
13-14
ATC
Use info* to identify special process for CVC3
* e.g. PAN, Discretionary data...
178
152
PAN 5432...0001
ATC 0004 > UN = 65654236
ATC 0005 > UN = 45743674
PAN 5432...0002
ATC 0005 > UN = 65727256
ATC 0006 > UN = 34246635
Cloud System
Issuer System
106

FIG. 15

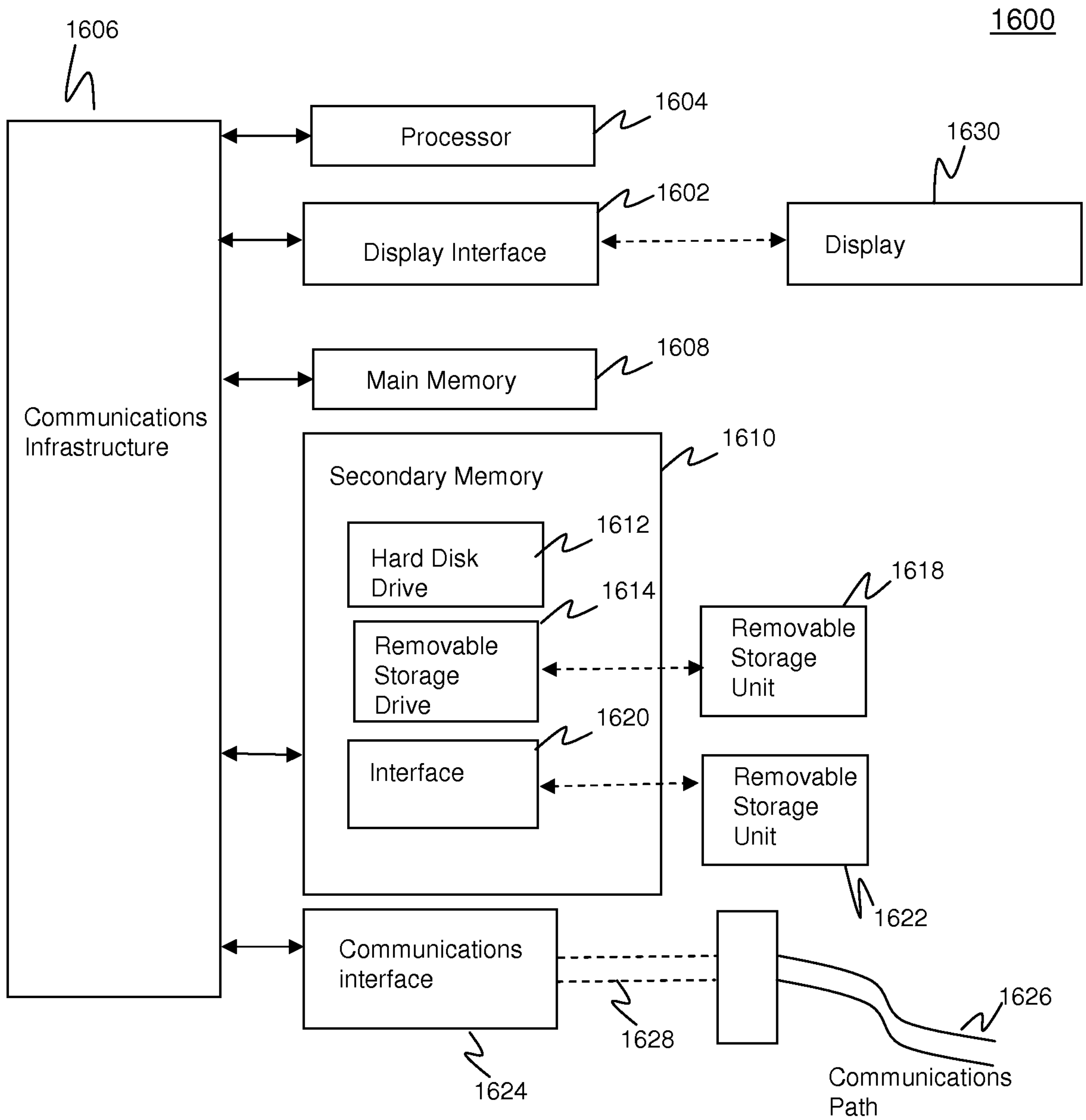

1600
1606
Communications Infrastructure
Processor
1604
Display Interface
1602
Display
1630
Main Memory
1608
Secondary Memory
1610
Hard Disk Drive
1612
Removable Storage Drive
1614
Interface
1620
Removable Storage Unit
1618
Removable Storage Unit
1622
Communications interface
1624
1628
1626
Communications Path

FIG. 16