MPIW Security Workgroup Initiative
Progress to Date and Current Status

Susan Pandy, Federal Reserve Bank of Boston
May 1, 2014

Susan Pandy is a Director in the Payments Strategies Group at the Federal Reserve Bank of Boston.

The views expressed in this paper are solely those of the author and do not reflect official positions of the Federal Reserve Banks of Atlanta or Boston or the Federal Reserve System. The author would like to thank members of the MPIW Security Workgroup for their thoughtful comments and review of the report.

Table of Contents

I. Overview

The security of mobile payments at the point-of-sale (POS) is a critical component of successful adoption. As new technologies, platforms, and solutions for mobile POS payments enter the market, efforts to understand and manage security become more complicated. Following a January 2013 meeting of the Mobile Payments Industry Workgroup (MPIW) facilitated by the Federal Reserve Banks of Boston and Atlanta, a sub-group of members known as the MPIW Security Workgroup was initiated to focus on mobile security and risk for retail payments at the POS.

The primary objectives of the group are to:

- Evaluate risks and security options of the different mobile platforms in the end-to-end[1] mobile payment transaction process;
- Determine which technologies and solutions should be part of the optimum mobile security environment;
- Recommend actions to ultimately improve the security profile and perception of mobile payments; and
- Enhance the payment industry and overall market understanding of the security elements of the mobile payments environment.

The scope of the project included an analysis of the security environment for proximity POS mobile payments. The group selected several mobile use cases that employ near field communication (NFC),[2] cloud,[3] and/or QR code[4] technologies to document end-to-end workflows. Next, the group conducted a threat analysis to identify the vulnerabilities in the transaction flows. They are now in the process of outlining possible mitigation tools and controls for each use case.

The deliverables to be completed in 2014 include:

- Validation of a use case analysis by the entire MPIW membership;
- Development of a risk model(s) that supports a secure solution for mobile phone payments at all points in the process, with a focus on authentication and identification of incentives for managing risk; and
- Completion of a final report with documented use case flows, and an evaluation of risks and recommended actions. Solutions may include mitigating actions by ecosystem stakeholders, awareness, risk acceptance, or referrals to appropriate standards bodies or policymakers.

[1] The end-to-end process begins with the consumer initiating payment and ends when the merchant or payee receives payment. The touch points in between include hardware, software, data transmission, and interfaces.

[2] NFC (near field communication): A standards-based wireless communication technology that allows data to be exchanged between devices that are a few centimeters apart. NFC-enabled mobile phones incorporate a smart chip (secure element) that allows the phone to store the payment app and consumer account information securely and use the information as a virtual payment card. NFC payment transactions between a mobile phone and a POS terminal use the standard ISO/IEC 14443 communication protocol also used by EMV and U.S. contactless credit and debit cards that allows the mobile phone to emulate a physical contactless card.

[3] Cloud is a remote server where payment credentials are stored and used to authenticate the payment transaction, instead of on the actual mobile phone. The cloud may be managed by a merchant or payment services provider.

[4] Quick Response (QR) code or 2D barcode is a two-dimensional barcode which contents can be scanned and decoded quickly.

II. Use Case Development

To more realistically describe each end-to-end transaction process, the workgroup developed use cases based on existing mobile payment solutions in the U.S. that employ NFC, cloud, and/or QR code technologies. Examples included in this report are: Isis, Google, LevelUp, Paydiant, and PayPal. Phase One included NFC/hybrid mobile payment models and Phase Two addressed cloud-based and QR code-based mobile payment models. (Note: In 2014, the group plans to evaluate recently announced models, such as host card emulation (HCE)[5] used in the latest version of Google Wallet.)

Use case analysis included identification of relevant parties to the transaction, hardware and/or software components, payment networks and other interfaces in the payment transaction flow, and authorization and authentication. The group also evaluated how the payment information for each transaction was transmitted and stored to identify potential threats and vulnerabilities at each point in the transaction chain, as well as to consider the scalability of those threats (i.e., low, medium, or high probability).

III. NFC Use Cases

The group began by mapping the two predominant NFC mobile payment scenarios in the U.S. market: the Isis Mobile Wallet and the Google Wallet. The Federal Reserve Bank of Boston discussed the distinction between a mobile wallet and a digital wallet[6] in its 2012 publication, *Mobile Phone Technology: "Smarter" Than We Thought*, noting that a mobile wallet describes the Isis model, which stores the payment credentials or actionable payment information on the phone in a SIM/UICC-based secure element (SE).[7] The SIM card/UICC (SIM) is proprietary, or owned by the mobile network operators (MNOs) that are part of Isis (AT&T, T-Mobile, and Verizon). The Google Wallet, while first introduced as a mobile wallet with an embedded SE (eSE) owned by the original equipment manufacturer (OEM), modified its solution to a hybrid model that allowed payment credentials to be stored in the eSE on the mobile phone *or* in the cloud. Its latest model uses HCE, in which the payment credentials are stored in the Google Wallet application data in its newest mobile operating system (OS) known as KitKat.

In Isis and the original Google model, the payment occurs via payment card emulation (i.e., the NFC–enabled mobile phone acts as a contactless card), with the payment credentials (e.g., Visa, MasterCard, etc.) stored in an SE in the mobile phone (i.e. hardware-based security). The SE essentially replaces the magnetic stripe on payment cards and stores encrypted payment card information that is transmitted over

[5] **Host Card Emulation** makes it possible to perform NFC card emulation without using the secure element (SE) in mobile handsets. HCE enables NFC card emulation communications to be routed through the mobile phone's host processor versus from the POS terminal through the NFC controller to the SE. The virtual/proxy prepaid card credentials are no longer stored in the tamper-resistant SE, but in the Google Wallet application data.

[6] Simply put, a mobile wallet is an application that controls access to and communicates with payment credentials stored in a secure element (physical chip) in the mobile phone to facilitate payment; a digital wallet is an application stored in the mobile phone that accesses payment credentials stored in the cloud (i.e., a remote server) via a mobile app, phone number and PIN or using a physical card.

[7] GlobalPlatform defines a secure element (SE) as a tamper-resistant one-chip secure microcontroller capable of securely hosting applications and their confidential and cryptographic data (e.g. key management) in accordance with the rules and security requirements set forth by well-identified trusted authorities. In payment applications, the SE controls interactions between trusted sources (bank), trusted application (mobile payment app) stored on the SE and third parties (company the user paying). The secure domain protects the user's credentials and processes the payment transaction in a trusted environment. There are three types of SEs—Subscriber Identity Module (SIM)/Universal Integrated Circuit Card (UICC), micro SD, and embedded secure element (eSE). The micro SD is not in the scope of this project.

a secure channel to retailers' NFC-enabled POS terminals.[8] Furthermore, each model involves three major steps in the end-to-end process: 1) enrollment, 2) provisioning and personalization of the payment credentials to the SE, and 3) the payment transaction. For an overview of a general NFC retail transaction flow, please refer to Appendix I.

Isis Mobile Wallet

Isis[9] is a mobile commerce joint venture created by three of the four largest mobile carriers[10] in the U.S.: AT&T Mobility, T-Mobile USA and Verizon Wireless. *(Sprint partnered with Google.)* Isis is not a financial institution (FI), but rather a service provider to FIs which enables mobile payment for the FI's payment accounts through the Isis platform and the Isis Mobile Wallet. Isis also acts as a service provider to merchants and advertisers through its ability to enable mobile offers and loyalty services.

Isis began to take hold in the U.S. in October 2012 when three issuing banks American Express (AmEx), Capital One, and Chase — added their credit cards to the Isis Mobile Wallet as part of a two-city pilot in Salt Lake City, Utah and Austin, Texas. In November 2013, Isis launched commercial service nationally and currently supports direct hosting of credit cards from Amex, Chase, and Wells Fargo.

The Isis Mobile Wallet also supports the Serve prepaid card from AmEx. This card allows the consumer to use their Serve account as a stand-alone prepaid account that they can prefund using cash, ACH (a demand deposit account (DDA)), or a linked credit or debit card. Additionally, Serve supports the ability to perform a real-time load of funds from a linked credit or debit card when the available balance is insufficient to cover the requested transaction amount, and thus can operate with an effective zero balance. This additional functionality offers a "bridge" for consumers who wish to use credit or debit cards in the Isis Mobile Wallet that are not currently supported for direct hosting. Serve card members can also initiate person-to-person (P2P) money transfers to other Serve accounts and online bill payments from within the Isis Mobile Wallet.

The Isis Mobile Wallet enables consumers to make purchases by tapping their NFC-enabled phones at physical POS retailers (including open loop transit systems such as the Utah Transit Authority and Chicago Transit Authority), receive and redeem coupons, and present retailer loyalty credentials. Loyalty and coupon information may also be delivered via a 2D barcode or other visual codes. Payments made through Isis Mobile Wallet rely on the traditional payment networks to authorize, clear, and settle transactions. For payment credential and other sensitive information storage, Isis uses a removable "Isis-Ready" special SIM card with an integrated SE.

[8] NFC is designed to be compatible with existing contactless reader technologies (e.g., ISO 14443), which allows the mobile phone to be tapped/waved at an NFC-enabled POS terminal to complete a contactless payment transaction.

[9] Its legal name is JVL Ventures, LLC, d.b.a. Isis.

[10] Based on the number of subscribers, the MNOs collectively hold 70%. Pyramid Research (2013). *From Digital Content to M-Wallets: M-Payment Strategies for Operators: Case Studies and Analysis of MNO Opportunities and* Approaches. Retrieved from http://www.pyramidresearch.com/store/mPayment-strategies-for-operators.htm.

Consumer Access and Enrollment

To activate and use the Isis Mobile Wallet, consumers must have a current wireless account in good standing with AT&T, Verizon, or T-Mobile USA that includes both SMS support and a cellular data plan. The consumer's mobile phone must be "Isis-Ready," meaning it is NFC-capable (or made capable via an NFC-enabled case for an iPhone), has been approved by Isis, and contains an enhanced SIM card which contains an SE smart chip. If the Isis Mobile Wallet is not already preloaded on the consumer's mobile phone, the consumer may download the application from Google Play or the Apple App store. To activate the Isis Mobile Wallet, the consumer must agree to the Isis Terms of Service, create a User ID and password (used to activate the wallet and log-in to the Isis website), and establish a 4-digit wallet PIN.

After the core Isis Mobile Wallet registration process is complete, the consumer may add one or more payment cards to the wallet. The current list of available bank-issued credit cards are displayed in the Isis wallet menu: AmEx, Chase, and Wells Fargo, as well as the AmEx Serve prepaid card. To add cards to Isis, the consumer enters his PIN to open the Isis Mobile Wallet and selects the bank from which he wants to add a card, via the wallet menu. Upon selection of a bank, the consumer is securely transferred from the Isis Mobile Wallet to a bank-operated mobile servicing website where consumer verification is performed and eligibility to load cards into the Isis Mobile Wallet is determined.[11] Once the bank/consumer interaction is successfully completed, the bank initiates the secure provisioning of the payment card into the Isis Mobile Wallet. Sensitive account information (e.g. account number, encryption keys) is ultimately stored in the SE.

Under this model, Isis has no visibility of consumer's payment card credentials or bank authentication information. Isis cannot access the account credentials in the SE; only the bank controls the encryption keys used to transmit account information to its domain within the SE. The cardholder's name is never stored in the Isis Mobile Wallet or in the payment applications on the SE and is therefore, never sent to the POS terminal during a transaction.

After enrolling, Isis Mobile Wallet users can use their mobile phones to make payments anywhere that contactless payments are accepted, globally. Any POS merchant with a contactless payment terminal can enable NFC subject to the normal acceptance rules and procedures of the card payment networks.

Role of Trusted Service Manager (TSM)

Isis utilizes a Trusted Service Manager architecture as described in the Global Platform[12] standards. This approach is also endorsed by the Groupe Speciale Mobile Association (GSMA).[13] A Trusted Service Manager (TSM) facilitates the secure loading and maintenance of sensitive information (in this case, payment account information provided by an issuer) to be stored in an SE resident on the consumer's SIM

[11] Consumer verification methods vary by bank.

[12] GlobalPlatform is a cross industry, non-profit association which identifies, develops, and publishes specifications that promote the secure and interoperable deployment and management of multiple applications on secure chip technology. For more information, see http://www.globalplatform.org.

[13] The GSMA represents the interests of mobile operators worldwide. Spanning more than 220 countries, the GSMA unites nearly 800 of the world's mobile operators with 250 companies in the broader mobile ecosystem, including handset and device makers, software companies, equipment providers and Internet companies, as well as organizations in industry sectors such as financial services, healthcare, media, transport, and utilities.

card. Isis partners with Gemalto to manage the Isis TSM services. Participating banks can contract separately for their own TSMs, either outsourcing or operating in-house, to interface with the Isis TSM.

The TSMs, SEs, and mobile devices used in conjunction with the Isis service undergo a complex and standardized process for testing, approval, and certification. Such tests and certifications are conducted under the guidelines, procedures, and standards defined by EMVCo, AmEx, MasterCard, and Visa. In addition, each MNO conducts its own battery of tests and certifications.

Payment Transaction Flow

During the mobile transaction process, the NFC controller in the mobile phone securely manages contactless communication between the mobile phone, the SE, the Isis Mobile Wallet client, and the merchant's POS device. The Isis Mobile Wallet client provides the user interface that allows the consumer to select which payment card to use for a transaction. The Isis Mobile Wallet provides the consumer with a confirmation screen when the mobile phone has transmitted the payment information.

To make an Isis NFC purchase, the consumer must first unlock their mobile phone (if it is locked) and then enter their Isis Mobile Wallet PIN to open the wallet. The consumer then selects one of the payment cards presented in the "card carousel" graphical user interface (GUI).[14] The consumer then holds the back of their mobile phone against the NFC-enabled POS terminal to establish a connection between the mobile phone and the terminal, which then transmits the selected payment card information.

On the back-end of the transaction, an Isis management application on the SE validates the Isis Mobile Wallet PIN and enables access to the selected payment card for that wallet transaction session. The Proximity Payments System Environment (PPSE) is populated with the Application Identifier (AI) associated with the specific payment card selected by the consumer. The POS terminal reads the PPSE file on the SE to determine which payment applications are available on the SE to perform the requested transaction. The POS terminal then selects the payment application instance and follows specific logic which is specified by the bank card issuer in accordance with the product requirements defined by the card payment networks.

All Isis supported payment applications make use of dynamic authentication data that is generated using issuer controlled encryption keys that were securely personalized during the card loading process and that change for each transaction. The POS terminal reads the payment account data and the dynamic authentication data needed for the transaction and transmits it to the payment card issuer via the normal payment card authorization procedures (i.e., via the merchant's acquirer/processor and the card payment network switch). The issuing bank completes the authorization process by verifying that the payment card account is legitimate, not reported lost or stolen, and has sufficient credit/funds available to pay for the transaction. In addition, it verifies that the dynamic authentication data is valid for that transaction. If approved, the issuer generates an authorization code and routes it back to the payment card network, which sends the code back to the acquirer. The acquirer sends the authorization decision code to the merchant POS terminal, which displays the approved or declined transaction status. If the transaction

[14] Entry of the Isis Mobile Wallet PIN authorizes use of payment cards within the Isis Mobile Wallet for a designated period of time which can be configured by the consumer from within the "Settings" menu in the wallet (Settings range from 1 minute to 30 minutes).

authorization is approved, the POS terminal will complete the transaction and provide a receipt to the consumer.

Isis Mobile Wallet Security

Each MNO owns and is responsible for the security of the NFC process and SIM-based SEs provisioned in mobile phones which are certified for use with the Isis Mobile Wallet. The SIM cards are subject to the same security testing and standards that traditional bank smart cards (contact or contactless) undergo prior to being issued. This includes both functional testing of the chip hardware, operating system (OS), and payment applications by AmEx, MasterCard and Visa; security testing of the chip hardware and OS (by EMVCo); and payment applications (by AmEx, MasterCard and Visa). Isis is responsible for the wallet service which uses the security features embedded in the mobile phone by the MNOs (i.e., the SE).

To prevent unauthorized access, Isis may lock the wallet application if the consumer attempts to login with an invalid wallet PIN, requiring the consumer to reset the PIN by re-entering the user ID, Isis password, and security answers. The Isis Mobile Wallet can also be suspended remotely by the consumer via the Isis website or by calling his/her MNO or card issuer. Under a similar process, if the consumer's mobile phone is lost or stolen, sensitive account information can be wiped from the wallet if it is within cellular coverage. Consumers are responsible for notifying their wireless service provider immediately of unauthorized use of Isis services, user ID, Isis password, wallet PIN, or loss or theft of the Isis-enabled mobile phone. Consumers are also instructed to contact their participating issuers to prevent unauthorized access to payment cards, or to restore access to them on another Isis-enabled mobile phone.

Google Wallet

The first version of the Google Wallet was launched in early 2011, and used an embedded secure element (eSE)[15] in the mobile phone. This model was based on the MasterCard PayPass contactless credit card. The payment card credentials were encrypted and stored in the SE in the mobile phone. For security purposes, the SE is separate from the Android phone's memory and designed to allow only trusted applications to access the payment credentials in the SE.

Version 2.0 of the Google Wallet was released in August 2012 as a hybrid NFC/cloud-based model. This version added a virtual prepaid MasterCard debit card issued by Bancorp to enable consumers to add any major credit or debit cards to Google Wallet, regardless of the bank card issuer. The virtual/proxy prepaid card was stored in the eSE in the mobile phone, allowing for a distinct "card" for each instance of a wallet. The virtual card linked to the actual (backing) payment instruments was maintained in Google's proprietary cloud.[16] Only one backing payment instrument could be active at a time.

The current model of the Google Wallet, version 3.0 (v3.0), was announced in October 2013 and shifted the wallet to a new Android platform capability known as HCE introduced in the Android KitKat (newest mobile OS version 4.4). HCE marked a shift away from previous versions of the Google Wallet that relied on the eSE in the mobile phone for the storage of the virtual/proxy prepaid card credentials. Instead, the new OS version stores the payment card credentials in the user data partition of the Android

[15] An embedded SE is built into the mobile phone and cannot be removed by the customer, unlike a SIM-based SE.
[16] Users had to provide the CVC2 code. The CVV2/CVC2 is a three-digit security code that is printed on the back of credit cards, to verify possession of the physical card.

KitKat OS, which leverages the OS-provided application sandboxing for security. These payment credentials are issued by Bancorp, provisioned to the mobile phone by Google, are only valid for a single transaction, and are automatically rotated after each transaction and after a certain period of time.

HCE was developed as a core capability of the Android OS such that non-Google payment applications can also make use of it. If there are multiple payment applications that support HCE on the mobile phone, the user can then select which payment application to use in the Android System Settings.

Enrollment and Payment Processing for Google Wallet v2.0

Before a consumer could enroll in the previous version 2.0 of the Google Wallet, they had to first purchase a Google Android certified/approved NFC mobile phone with an eSE and then download the mobile wallet application (app) from the Google Play or Apple app store.[17] Once the app was installed on the mobile phone, the consumer would then be prompted to enter his 5-digit U.S. zip code and to opt-in to receive updates. The consumer would then create a four-digit wallet PIN, which would also be used to perform verification for debit transactions and cash withdrawals using the Google Wallet payment card.

To initiate a payment transaction, the consumer would need to enter a PIN to open his mobile phone (optional). He then tapped the mobile phone at the NFC-enabled POS reader/terminal at a merchant location. The payment applet for the selected payment choice would contact the NFC controller and prompt the user to enter his four-digit wallet PIN (required) to authorize the transaction. The NFC controller would send the PIN to the eSE for verification by a Mobile MasterCard PayPass (MMPP) applet, which unlocked the payment applet. The MMPP applet authenticated the user and transmitted the encrypted virtual proxy account number from the eSE through the POS terminal to the cloud to request authorization of payment from the selected credit card network.[18] The authorization code was routed back to the cloud, which communicated back to the POS terminal. The consumer either received a confirmation of payment either from the NFC POS terminal or the mobile phone screen would signal a visual or audible confirmation of the completed transaction. The consumer would also get an email receipt of his purchase. Settlement of the virtual card payment was handled between Bancorp and the merchant.

The merchant did not receive the actual credit or debit card credentials in this transaction. Google Wallet facilitated payment to the merchant for the in-store purchase using the virtual prepaid MasterCard payment card, treating the payment as **card present to the merchant**. Google then used the backing instrument to charge the purchase to the selected debit or credit card network, which treated that transaction as **card-not-present (CNP)**. Google was the issuer for the proxy virtual payment card and the merchant of record for charges and chargebacks placed on the real backing instruments. Authorization requests on the virtual card were approved only after successful authorization for the identical amount on the backing instrument in the cloud. No credit lines were extended to users. Transactions with linked cards appeared on consumer bank statements as "GOOGLENFC* [Merchant name]". Even though the consumer physically presented his mobile phone containing the virtual prepaid card at the POS and received authorization approval from the backing instrument, the card network still considered the transaction CNP.

[17] The customer must already have a Google account to access the Google Wallet app.

[18] Transactions are considered "paying with credit."

Enrollment and Payment Processing for Google Wallet v3.0

Google Wallet (v3.0) was introduced in October 2013 and built on the latest version of its mobile OS system, Android 4.4 KitKat with its new HCE feature. HCE enables NFC transactions at the POS in card emulation mode but rather than the information being routed from the POS terminal to the NFC controller to the SE in the mobile phone, it is routed from the NFC controller to the mobile phone's host processor. In effect, HCE introduces a routing mechanism that can support multiple wallet and non-wallet applications that leverage the cloud.

Enrollment for Google Wallet 3.0 basically follows the same process as previous versions. However, sthe updated OS KitKat 4.4 will prompt the consumer to approve "tap-and-pay" payments for Google Wallet. Tap-and-pay payments are only available to consumers with KitKat 4.4 or higher on their mobile phones. Supported devices will display a tile in the consumer's "My Wallet" screen that instructs them to set up tap-and-pay. For devices that are not eligible for Android 4.4 KitKat or that do not support tap-and-pay, consumers can still use the Google Wallet app to store all of their loyalty cards and offers, send money, view orders, and use the Google Wallet payment card to make purchases.

To pay at the POS, a user unlocks their handset and taps. If the user has not recently entered a PIN, Google Wallet will prompt the user for one. Entering a PIN requires a network connection. Upon completion of the transaction, a transaction notification appears on the mobile phone screen for the consumer.

During a payment transaction, the NFC controller in the mobile phone emulates the passive-mode behaviour of a smart card rather than being an active-mode NFC participant. The Android OS dispatches the calls to the Google Wallet application for handling, activating the Google Wallet if it is not already running**.**

Potential Vulnerabilities in NFC Contactless Payment Models

Since NFC facilitates contactless transfer of information, it is exposed to certain security risks as discussed below, including eavesdropping, denial-of-service attacks (DoS), data manipulation or corruption, and relay attacks. It should be noted that the risks should be considered in the context that the range of NFC devices is limited – usually only a few centimeters.

Eavesdropping occurs when a third party intercepts a transmission between the mobile phone and the POS terminal and gains access to the data being transmitted.[19] If the data is sensitive, such as payment card data or personal information, then the third party will have full access to this data and can perpetrate fraud or clone the payment card credentials. Because eavesdropping can be prevented by using encryption methods to establish a secure channel between the NFC devices, and because CVC3 deters cloning from simple eavesdropping, this is considered a low risk, low probability event. And unlike plastic credit cards, contactless card track data does not contain the cardholder's name.

A denial-of-service attack (DoS) is an explicit attempt by an attacker to prevent legitimate parties from using a particular service. DoS attacks include "flooding" the merchant's transaction processing network

[19] Interception may be carried out using an improvised antenna within 20-30cm. Because the transmission range is so short, NFC-enabled transactions are proclaimed to be inherently secure.

with messages to prevent legitimate data traffic; or disrupting connections between the POS terminal and servers. Such attacks usually disrupt the merchant's or other party's services for malicious rather than fraudulent reasons, although there have been cases where the DoS attack was initiated to serve as a distraction while other network intrusions generated illegal transactions. A DoS attack usually disables the entire merchant POS system and not merely the mobile payment transaction.

Data corruption or modification is a type of DoS attack. It occurs when a third party intercepts the data transmission from the mobile phone and alters or disrupts the data before sending it to the receiving party (e.g., POS terminal), which prevents completion of the transaction. This type of attack is very difficult to implement, but may occur in rare cases, especially for active mode[20] transmission of NFC information.[21] Mitigation efforts include: 1) the sending device checking the radio frequency (RF) signal during data transmission to detect the attack and stop the data transmission automatically,[22] or 2) using a secure (encrypted) communication channel.

A relay attack occurs when an attacker physically places a second "proxy" NFC reader near the victim's NFC-enabled phone to intercept the communication, modify or record the data, and then relay the data to the legitimate NFC reader. The communication range must be in close proximity (i.e., ISO/IEC14443[23] protocol compliance)[24] so it is difficult to achieve this attack with NFC. Mobile relay attacks can be prevented by using active-passive communication mode which enables the NFC phone to hear and detect an unwanted third party using a secure communication channel.

Security Controls for NFC Contactless Mobile Payments

Secure element: All payment credentials are stored in the SIM or embedded secure element (eSE). The combination of NFC and an SE to protect mobile payments between the mobile phone and the POS is considered tamper-resistant.[25] Prior to installation in the mobile phone, the SE goes through extensive testing and certification processes. Once contained in the mobile phone, the SE requires authenticated access rights because the SE is secured from the rest of the operating system (OS). The SE is encrypted to prevent extraction and duplication of payment card data inside. Only a TSM that shares cryptographic keys with the SE can perform certain operations, such as modifying the payment applets installed on the SE. If malware successfully penetrates a mobile phone, the NFC controller is able to block the malware request to the SE because it recognizes the malware as an unauthorized application.

---

[20] In active mode, two active units communicate with each other, vs. passive mode where the communication is one-way.

[21] For more information, see Haselsteiner, E. and Breitfuß, *Security in Near Field Communication (NFC) – Strengths and Weaknesses*. Retrieved from http://ece.wpi.edu/~dchasaki/papers/Security%20in%20NFC.pdf.

[22] This is feasible because the power to corrupt data is bigger than the power used to send normal data and would be detected.

[23] **ISO/IEC 14443** *Identification cards -- Contactless integrated circuit cards -- Proximity cards* is an international standard that defines proximity cards used for identification, and the transmission protocols for communicating with it. For more information, see http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39693.

[24] See http://pure.rhul.ac.uk/portal/en/publications/practical-relay-attack-on-contactless-transactions-by-using-nfc-mobile-phones(158a0a34-227a-4807-952d-e3a07fa7052a).html.

[25] The secure element (SE) is programmable, but not an open platform where any arbitrary code can be installed. Each SE is assigned unique cryptographic keys for payment protocols that are stored in the SE for the TSM to use to provision new applets OTA, which are managed by the wallet application. Tamper-resistant: The SE is highly tested and protected from tampering at the hardware level. Because of such strict production controls, it is highly unlikely that the SE could be tampered with during production or shipping to the MNO, TSM or service provider for provisioning. The tight security makes it best suited for security-critical functions requiring encryption, such as payments. The wallet application (Google or Isis) then manages the applets and payments.

The cryptographic keys for the payment credentials are stored in the certified, tamper-resistant SE and never revealed to the merchant. The mobile OS configuration, set by the Original Equipment Manufacturer (OEM) or MNO, controls access to the SE. The host OS cannot access the keys directly. A customer 4-digit PIN is required for authentication to activate NFC after opening the mobile wallet application, in addition to an optional PIN to unlock the actual phone. The SE is only accessible when the mobile phone screen is on and locks after six incorrect attempts to access it. The payment card credentials, as required by standards, are encrypted as they are transmitted wirelessly from the mobile phone to the POS terminal. The NFC signal must be in close proximity to an NFC reader to be received, and the antenna shuts off automatically, once the transaction completes.

There are many types of rules, standards, and agreements that contribute to the security of the NFC payment in the SE.[26] GlobalPlatform's secure channel protocol provides for transmission of sensitive account data between the TSM and the SE and storage of the data in the SE. MNO-provided encryption further secures account credentials against over-the-air (OTA) sniffing.

NFC-based mobile payments also allow for the use of dynamic data authentication (DDA) to generate unique, one-time transaction-specific card verification (CVV) data (dynamic vs. static data on magnetic stripe) to protect the mobile payment transaction.[27] DDA reduces fraud because transactions can only be generated with a genuine payment card (or from a card properly encoded on a mobile phone SE), and cannot be replicated or cloned.

GlobalPlatform (GP) is developing the Trusted Execution Environment (TEE),[28] a new framework for mobile phone security. The TEE is intended to provide a higher level of protection from software attacks than the rich OS. GP is analyzing how the TEE will access the SE and what role the TEE will perform in the POS payment ecosystem.

Regardless of the mobile phone technology, there are tools to protect against a lost or stolen mobile phone and unauthorized access. Isis and Google Wallets require customers to enter PINs to unlock their wallets. Customers may also remotely disable the payment card credentials stored in their wallets to avoid fraudulent use and request that the wallet application be remotely wiped from their mobile phones. After completing an NFC mobile wallet transaction, a consumer can turn off the NFC transmission capability and close the payment applet on his mobile phone for further protection.

Table 1 shows examples of threats, probability of vulnerability, and some known mitigations and controls for the three types of secure elements: SIM/UICC, embedded SE (eSE), and microSD. This table is a very high level, preliminary analysis that will be further developed in Phase II of the workgroup project.

---

[26] NFC and SEs have multiple international standards associated with them, including ISO 14443 (for contactless integrated circuit cards), 18092 (for NFC interface and protocol for wireless connections), 13157 (for telecomm NFC security), etc. The NFC Forum provides multiple specifications that must be followed, as does Global Platform, GSMA, SIM Alliance, ETSI, and EMVCo. Based on all these NFC-related standards, there is a strict process for controlling, testing and certifying NFC and SEs before they are provisioned to phones for use. The certification process can differ slightly depending on the payment card network. Smart Card Alliance (2012, November). *Mobile/NFC Standards Landscape Reference Guide*, retrieved from http://www.smartcardalliance.org/resources/pdf/Standards_Reference_Guide_FINAL_103112.pdf.

[27] The mobile phone and NFC function do not generate the dynamic data authentication, but the payment applet in the SE generally requires the use of DDA under EMVCo. Dynamic data authentication is a means to make the chip nearly impossible to counterfeit. In EMV transactions using DDA, the chip is a mini computer that generates a unique cryptogram using transaction data each time the card is inserted into a chip terminal. The cryptogram is then sent to the card issuer, which uses its keys and codes to calculate a cryptogram based on the same transaction data.

[28] The TEE will be further analyzed by the workgroup in 2014.

Table 1. Secure Element Threats, Vulnerabilities, Mitigations & Controls

Secure Element Type	Threat	Vulnerability (Low, Medium, High)	Mitigations/Controls
SIM/UICC	Cloning	Low	• Conform to ISO 7816 – Organization, security and commands for interchange • Conform to ISO 14443 – ID Cards – contactless integrated circuit cards – proximity cards • Conform to GP smart card specifications/compliance program • Conform to ETSI Smart Card Platform specifications • Conform to GSMA's Security Accreditation Scheme (SAS)[29] • Conform to mobile communication standards 3GPP TS 11.11 http://www.3gpp.org/ftp/Specs/html-info/1111.htm • Single Wire Protocol • EMVCo testing, security evaluation[30] • Standards regarding physical characteristics and electrical interface • Card network customized testing and security evaluations of SIM to certify payment app prior to release
	Replay Attacks[31]	Low	• Stronger controls around designated apps that access the SE
	Pre-play Attacks[32]	Low	• Stronger controls around designated apps that access the SE
MicroSD	Removable/ portability	Low-Medium	• Conform to ISO 7816 – Organization, security, and commands for interchange • Conform to ISO 14443 – ID Cards – contactless integrated circuit cards – proximity cards • Conform to Global Platform smart card specifications/compliance program • SD association[33] standards for memory cards

[29] SAS is a voluntary scheme through which UICC suppliers subject their production sites and processes to a comprehensive security audit by GSM operators. Successful sites are awarded security accreditation for a period of two years. For more information, see http://www.gsma.com/technicalprojects/fraud-security/security-accreditation-scheme.

[30] EMVCo, after it receives a letter of qualification that an SE has met all GlobalPlatform core requirements, conducts its own test of the SE application activation user interface to ensure that the end user can easily select his preferred payment provider via the mobile handset screen. EMVCo also certifies that the SE meets the required payment security standards (e.g., PCI DSS).

[31] In a replay attack, the phone's Internet connection is used to receive and execute commands sent by another remote phone, enabling the remote device to emulate the SE of the target device without physical proximity.

[32] In a pre-play attack, if the attacker is able to physically collect and analyze transactions (e.g., by infecting a terminal (ATM or POS) with malware, or by a man-in-the-middle attack between the terminal and the acquirer, that sends the data remotely), he can save the authentication data from a particular time and re-use it at a later time pre-determined by the counter. In effect, pre-play attacks allow criminals to send fraudulent transaction requests from rogue chip-enabled credit cards. See Bond et al. (2012). [Working Paper] *Chip and skim: cloning EMV cards with the pre-play attack.* University of Cambridge, UK, accessed from http://www.cl.cam.ac.uk/~rja14/Papers/unattack.pdf.

			• SD association supports ETSI Single Wire Protocol on the microSD.
Embedded SE (eSE)	Cloning	Low	• Conform to ISO 7816 – Organization, security and commands for interchange • Conform to ISO 14443 – ID Cards – contactless integrated circuit cards – proximity cards • Conform to Global Platform smart card specifications/compliance program
	Replay Attack	Low	• Stronger controls around designated apps that access the SE
	Pre-play Attack	Low	• Stronger controls around designated apps that access the SE

IV. Cloud-based Use Cases

Cloud technology supports digital wallet mobile contactless payments. Cloud computing usually refers to the delivery of computer processing infrastructure, applications, and remote data storage facilities over the Internet. Some mobile solution providers use cloud technology to store financial payment credentials as an alternative to NFC SEs in the mobile phone. Deployment models include private, community, public, and hybrid,[34] but the examples of digital wallet solutions that follow use either private (proprietary) or public/private clouds (private services provided by a cloud service provider to multiple tenants for a particular industry).

For payments, the prevailing cloud model is *card on file*, in which the payment provider stores the consumer's credit and/or debit card credentials in a remote file server (cloud)/digital wallet instead of in the SE in the mobile phone. The credentials are stored either in a tokenized and/or encrypted format, discussed in the LevelUp, Paydiant, and PayPal examples.[35]

LevelUp

LevelUp, launched in March 2011, is a cloud-based, closed-loop digital wallet with a mobile application (app). Only participating merchants and consumers can use LevelUp. It uses a triple tokenization[36] process to protect payment credentials. Triple tokenization requires a combination of tokens (consumer QR, LevelUp, and Braintree) and two other authentication factors (consumer PIN to access LevelUp app and mobile device ID[37]) to initiate a transaction. This allows LevelUp to eliminate the need to share or

[33] The SD Association is a global ecosystem of companies setting industry-leading memory card standards that simplify the use and extend the life of consumer electronics, including mobile phones. The Association creates standards and promotes the adoption, advancement, and use of SD standards used by manufacturers of interoperable memory cards and devices. For more information, see https://www.sdcard.org/home/.

[34] For more information, see http://www.cloudtweaks.com/2012/07/the-4-primary-cloud-deployment-models/.

[35] Since PayPal can also tie a consumer's PayPal account to their bank account (ACH), PayPal also uses an *account on file* model.

[36] Tokenization creates a proxy set of identifying information for the real payment card data, eliminating the need for merchants to transmit and store sensitive financial and personal information. Tokens can be processed in place of the original card credentials. Tokenization is a data protection tool comparable to encryption. Tokens have no value to an attacker if stolen or accidentally exposed.

[37] A device ID is distinctive number associated with a smartphone or mobile device.

store payment credentials with the merchant or within LevelUp servers. All financial data are encrypted during transmission and at rest.

The process for consumer enrollment, account creation, and making a LevelUp payment are detailed below.[38]

Enrollment and Account Creation

1. Consumer downloads and installs the LevelUp app on his mobile phone from the mobile app store (e.g., iPhone App Store, Google Play Store).
2. Consumer registers and creates an account with LevelUp by inputting his email address and creating a username and password in the mobile app. All data is transmitted over a secure, encrypted connection.
3. From within his authenticated account, the consumer can add a payment card for funding. LevelUp encrypts the cardholder information with a public key provided by Braintree,[39] for which LevelUp does not have the associated private (decryption) key. The consumer can enter his card credentials manually or use his mobile phone camera to take a picture of the credit or debit card. The encrypted data passes through the LevelUp server to the Braintree server without the ability of LevelUp's platform server to access the underlying plain text cardholder information. This has the effect of keeping LevelUp's platform server out of the Payment Card Industry Data Security Standard (PCI DSS)[40] cardholder data environment.
4. Once Braintree receives the payment card credentials, it returns a token identifier (QR code) to LevelUp, which is stored in LevelUp's database.
5. LevelUp then creates a unique token for each consumer, which generates the QR code displayed on the consumer's mobile phone and scanned at the POS to pay for purchases. The consumer can reset this token at any time; it contains no payment account information.

Payment Transaction

To pay the merchant, the consumer opens the mobile app, logs in, and scans his personal LevelUp QR code at the LevelUp merchant terminal[41] which is a smartphone based in a LevelUp dock or a LevelUp scanner and connected to a standalone or POS integrated networked QR code reader. The merchant terminal reads the QR code data and translates it to a simple string of characters to generate a second token (i.e., a scan ID)[42] that validates the uniqueness of the scan. The cashier enters the purchase amount and submits the transaction. The consumer's QR code token, the scan ID token, and the merchant authentication token are encrypted and transmitted securely[43] to LevelUp's servers to be validated.

If validation is successful, LevelUp unlocks the Braintree's identifier token and securely transmits it with the payment amount and LevelUp's authentication token from LevelUp's servers to Braintree. Braintree

[38] See Appendix IV for an illustration of the transaction flow.
[39] Braintree is LevelUp's cloud processor for mobile payments. The Braintree server performs like a proprietary cloud with subsections and private clouds for each service. Braintree was recently acquired by eBay.
[40] See https://www.pcisecuritystandards.org/security_standards/documents.php?association=PA-DSS.
[41] The terminal is either a standalone networked reader or a POS integrated networked 2D barcode reader.
[42] The scan ID is a charge mechanism to detect and filter duplicate charges. It is generated by Akami, a LevelUp partner, which communicates with the POS terminal. When Akamai generates the unique charge identifier it attempts to forward the charge request to LevelUp's platform and backup servers, in that order. If the charge ends up in both places, the unique charge ID is used to filter out duplicates.
[43] Merchant terminal is transmitting over a secured connection to LevelUp's servers which are compliant with (PCI DSS).

uses the identifier token to validate the LevelUp authentication token. If successful, Braintree encrypts and transmits the real credit card data[44] to the issuing bank to handle the authorization and settlement. The bank processes the transaction and if all validations are met and the payment is approved, it passes back a confirmation to Braintree; otherwise it passes back a failure. Braintree returns a success/fail confirmation to LevelUp, which returns the confirmation to the merchant POS device and delivers a push notification and email receipt instantly to the consumer's mobile phone.

LevelUp Security

The LevelUp model protects the security of the consumer's payment credentials by never sharing or storing them with the merchant or on LevelUp's servers. Instead, Braintree handles the storage and transmission of consumer payment credentials and other sensitive data. The Braintree server performs like a proprietary cloud, with subsections and private clouds for each client, and provides a triple-based tokenization[45] process that protects the credentials. Triple-based tokenization requires a combination of tokens (the randomly generated consumer QR code, the second token on the LevelUp server, and the third token in the Braintree vault) and two other authentication factors (customer PIN to access LevelUp app and mobile device ID[46]) to initiate a transaction.

All financial information is encrypted during transmission and at rest. While a low probability, there is always the risk that other tenants/platforms could gain access to or impact other servers in Braintree's cloud system. But fraudsters would require significant internal help to break into these systems and the IP packet can be traced to where the hack originated (down to the country, neighborhood, etc.).

No personal information is sold to other third parties, including businesses that work with LevelUp. The company states that it is 100 percent compliant with all PCI DSS requirements.

Internal security controls deployed by LevelUp for the consumer include an instant QR code reset that can be done at any time from within the account online. This process generates a new QR code, cancels the old QR code, and de-authenticates all devices logged into the consumer's account. The consumer can then log back into his LevelUp account to access the new QR code. LevelUp also allows consumers to PIN-lock the LevelUp app by tapping "Settings" and entering a PIN-lock combination. The consumer also has the option to PIN-lock his mobile phone.

With each transaction, the consumer receives a digital receipt via email and push notification. If something appears wrong with a transaction, the consumer can reset his QR code with a single click to stop all activity on his account. Payment transactions made using LevelUp are identified with LevelUp and the individual merchant's name on a consumer's payment card or bank account statement.

[44] Braintree's servers have secure encrypted connections to the bank and to LevelUp.
[45] Tokenization creates a proxy set of identifying information for the real payment card data, eliminating the need for merchants to transmit and store sensitive financial and personal information. Tokens can be processed in place of the original card credentials. Tokenization is a data protection tool comparable to encryption. Tokens have no value to an attacker if stolen or accidentally exposed.
[46] A device ID is distinctive number associated with a smartphone or mobile device.

Paydiant

Paydiant offers a white label, cloud-based digital wallet solution that enables banks, retailers, and processors to deploy their own branded, contactless mobile wallet, mobile payment and cash access platform without involving intermediaries. Paydiant is a merchant acceptance network and does not interface directly with consumers. Merchants use Paydiant software to add payments, offers, and rewards capabilities to their proprietary applications (apps). Through Paydiant, merchants can accept PIN/signature debit, credit, prepaid, and loyalty cards for payment in multiple venues including POS stores, online, at the table (dining), or to pay bills online or from a paper invoice.

The consumer accesses the wallet application by enrolling for a digital wallet with a participating merchant or bank.[47] To register, the consumer downloads the bank- or merchant-branded mobile app and creates an account and a passcode. He then provides payment credentials for the accounts (credit, debit, checking account, prepaid) to be linked to the wallet application. During registration, the Paydiant system creates a customer identifier to uniquely distinguish the consumer to the bank or merchant.

Paydiant utilizes a transaction ID to substitute for the payment instrument to facilitate a mobile payment transaction. This method eliminates communication of sensitive data, even in tokenized form, between the mobile wallet and the POS terminal and allows the consumer to control the transaction from his mobile phone. Using a transaction ID also allows the consumer to pay in other venues such as e-commerce and paper bills, where traditional POS scanners are not available.

Payment Transaction

The steps in the Paydiant payment transaction flow are described below.[48]

1. Consumer enters his passcode to log-in to the merchant- or bank-branded Paydiant app. Paydiant server performs two-factor authentication by verifying the consumer's passcode and the mobile phone fingerprint (i.e., device ID).
2. Merchant totals the purchase and selects "mobile" as the tender type. POS terminal sends the transaction details and an authorization request to the Paydiant server.
3. Paydiant generates and transmits a unique transaction identifier (checkout token) representing the sale to the POS terminal where it is displayed as a QR code.[49]
4. Consumer scans the QR code with his mobile phone.
5. Mobile app sends QR code/transaction ID and payment request to Paydiant.
6. Paydiant communicates to mobile phone to ask consumer to select payment method from stored payment accounts, and any available offers.

[47] Because the Paydiant wallet is customized and branded for each particular bank or merchant, the consumer does not know that Paydiant is the wallet provider.
[48] See Appendix V for an illustration of the transaction flow.
[49] Paydiant is technology-agnostic in terms of how to present the transaction ID (e.g., QR code, NFC, Bluetooth Low Energy signal, etc.) because it drives everything through the cloud and its reverse transaction model. However, QR code presentment is the most popular deployment method among its partners because it is easy to display at the POS, and fast and inexpensive to deploy.

7. Consumer selects payment method and confirms the transaction. Mobile app sends the payment transaction and authorization request to the Paydiant server for processing with the appropriate payment card network and issuer.
8. Once authorization is approved, issuer/processor sends a response (assumed to be an approval) back to the payment card network, which sends it to Paydiant.
9. Paydiant sends the response to the merchant POS terminal and sends the consumer an e-receipt for the transaction.

Paydiant Security

The Paydiant process is referred to as a "reverse transaction flow" because it allows the consumer to capture the QR code and transmit the payment transaction request message directly to Paydiant's cloud server. Both Paydiant and the merchant have the ability to identify the consumer before a payment has been completed, allowing fraud tools to begin scoring the interaction before any funds are exchanged.

Paydiant also limits the movement of the payment credentials to a secure connection between Paydiant's platform and the merchant processor, which further reduces the potential for liability or fraud by not exposing actual or tokenized payment instruments to the POS. Using the transaction identifier/QR code that references a transaction stored in the cloud ensures that only the payment instrument associated with the mobile phone that captured the transaction token can be used to make a payment. These tokens are session-based and not visible.

PayPal

The PayPal digital wallet uses PayPal's proprietary cloud to allow users to pay from multiple accounts, store and use gift cards, access special offers, and store receipts. The digital wallet has several features including check-in with photo identification, authorization code/QR code, hands free, and Beacon.[50] To begin the process, the consumer enrolls online by creating a PayPal account, linking his funding methods (e.g., credit, debit or prepaid cards and/or bank account), and downloading the mobile app. If a consumer uses funds stored in his PayPal account to pay for a purchase, PayPal authorizes and settles the transaction between the consumer's and merchant's PayPal accounts within the PayPal system. PayPal only passes the transaction to a payment network if the consumer uses a payment card as his primary funding method.

Check-In with Photo Identification or QR Code

A consumer can activate Check-In through his PayPal mobile app. He must opt-in to allow the app to access the phone's geo-location data in order to let him locate participating (typically small) merchants. The consumer must also upload a digital photo that allows participating merchants to recognize the consumer when he enters their establishment and to verify the consumer's identity before completing a transaction. When a consumer enters a participating merchant location, the mobile app allows the consumer to check-in with that merchant and the merchant receives the consumer's digital photo to its POS system. The transaction process for PayPal Check-In is outlined below.[51]

[50] The handsfree and Beacon variations are not discussed in detail here, but will be reviewed further in the future. The handsfree method allows consumers to pay at PayPal-enabled terminals by typing in a PayPal PIN and their mobile phone number (does not require mobile phone to be present). Merchants do require a software upgrade to their terminals to accept PayPal. PayPal Beacon allows consumers who opt-in to be automatically checked in to a store upon entering using Bluetooth low energy.
[51] See Appendix VI for an illustration of the Check-In transaction flow.

1. Consumer opens his PayPal mobile app, enters his PIN, and selects the merchant store he is entering.
2. Check-in is established via a secure Internet channel to the PayPal proprietary cloud and communicated to the merchant POS device.
3. When ready to pay, consumer notifies cashier he is paying with PayPal to initiate the transaction.
4. Cashier selects "PayPal," opens *PayPal Here* mobile app, and logs in.
5. Cashier finds Check-In consumer by his photo, first name, and last initial and confirms that this is the right consumer.
6. Cashier enters amount to be charged, clicks "charge" and then "complete payment."
7. POS system transmits amount charged to the cloud provider which communicates a confirmation or decline of the transaction.
8. POS sends consumer confirmation and receipt based on pre-set preferences (e.g., email, text, or push notification).

Authorization/QR Code: PayPal also enables consumers to pay using a QR code (online token) instead of the photo identification. When ready to pay, the consumer opens the PayPal app and checks-in at the merchant location. The app prompts the consumer with a QR code to authenticate his purchase. If the merchant has a QR code scanner, the merchant scans the QR code to complete the payment. If not, then a four-digit code will display on the consumer's mobile phone for him or the merchant to key in.

PayPal Security

Sellers and merchants never see consumers' sensitive financial data because PayPal stores consumers' personal information on PayPal servers that are protected both physically and electronically. To provide an extra level of security for credit card and bank numbers, PayPal does not directly connect their firewall-protected servers to the Internet. PayPal also has an extensive risk management process with security specialists providing 24x7 monitoring of transactions for suspicious activity.

Email Confirmations: PayPal sends an immediate email confirmation of a transaction whenever a consumer sends or receives a PayPal payment. PayPal will launch an investigation if the consumer alerts them that he did not make the transaction.

Data Encryption: PayPal encrypts all sensitive information and email communications using the highest level of commercially available encryption (128-bit). When a consumer registers or logs in to PayPal, PayPal confirms that the consumer's browser is running Secure Socket Layer 3.0 (SSL) or higher, which secures the communication from the browser to Paypal.com.

All information in transit is protected by SSL, which ensures that the information is encrypted to prevent any theft during transmission over the internet. PayPal also has a security key that can be used as a second authentication factor when logging in to a PayPal account. The PayPal Security Key sends the consumer a One-Time PIN (OTP or temporary security code) via SMS to his mobile phone. This OTP is unique for each login. The consumer enters this code in addition to his password when logging into PayPal.

Potential Vulnerabilities in Cloud-Based Payment Models

The workgroup evaluated primary threats and vulnerabilities that can affect cloud-based payment models and will provide further analysis in Phase II of the project.[52] Some of the topics discussed included insider threats (e.g., from a cloud service provider employee with access rights to sensitive information), denial-of-service (DoS) attacks, data breach and compromise, and hardware threats to the consumer's mobile phone,[53] merchant POS terminals, and/or barcode readers.[54]

Cloud service providers can be primary high-value targets for these types of threats because they manage and store the payment card credentials and related data. To help mitigate threats, they must comply with PCI DSS requirements for secure storage of sensitive cardholder data and also conduct a risk analysis of sensitive information that is stored or processed by other parties to the transaction flow.

Security Controls for Cloud-Based Payment Models

Tokenization and end-to-end encryption (E2EE) are the primary methods to ensure the security of cloud–based payment models. The two techniques are often combined. Multiple methods exist for generating tokens and protecting the overall system, but, in contrast to encryption, no formal tokenization standards exist.[55] Often the distinction between encryption and tokenization is blurred, but both approaches involve the use of secrets to protect data and in both cases the data that is protected is only as safe as the transformation process itself and any secrets that support it. Just as encryption keys need to be managed in a highly secure way, the tokenization process and token data vault must also be highly protected.
The cloud-based payment models reviewed in this report all utilize a tokenization system and various methods of encryption.

V. Next Steps and Conclusion

The MPIW Security Workgroup will further explore the risk of cloud-based payment models and various securitization methods, particularly tokenization, end-to-end security, and point-to-point security in 2014. The group will incorporate this additional risk information into the use cases. We will also examine the security considerations and potential vulnerabilities of emerging technology models - such as host card emulation- and continue to evaluate the threats, vulnerabilities, mitigations, and controls, including assignment of probability associated with various risks and threats.

---

[52] The Cloud Security Alliance has identified the top seven threats to cloud computing in its 2010 *Top Threats to Cloud Computing V1.0*, available at https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf. While not specific to the payments industry, this resource is a good guide to understanding the prevailing threats in a cloud-based environment.

[53] The mobile device is particularly susceptible to a user's failure to protect the device with PINs or passwords. Users may also unknowingly download nefarious mobile apps that can introduce malware to the device that can impact digital wallet applications that reside on the mobile phone.

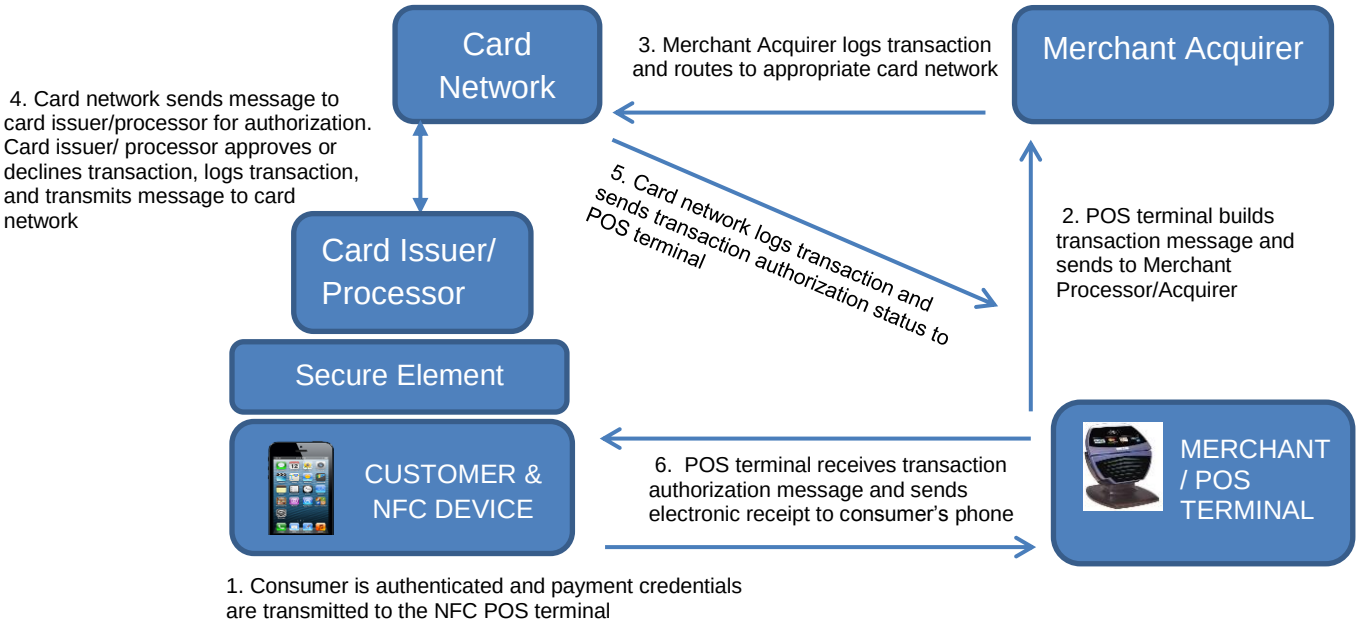
[54] The POS hardware – both terminals and barcode scanners–may be susceptible to fraud and hacking attacks. These threats tend to be higher among smaller merchants (i.e., less than 1 million transactions) who may not have adequate security controls in place, such as those required by PCI DSS.

[55] The Clearing House, a financial institution trade association, is developing a tokenization solution/standard that will rely on a secure cloud-based payment model in which the financial institutions collaborate to hold the credit card information on their servers. This model will be analyzed more in 2014 after it is implemented. American Express, Discover, MasterCard, and Visa have also announced a plan for standardizing tokens for use throughout the payment ecosystem which will be further analyzed in 2014.

Because the securitization behind NFC smart chips is not very well understood and might benefit from further analysis, we plan to discuss the complexities of NFC certification processes to identify opportunities for improvement and simplification. Given the lack of standards to support cloud technology, tokenization, and related security, the group will research current standards efforts and determine if there are gaps for which recommendations could be developed and submitted to recognized standards bodies.

The group's overall goal is to develop materials to educate consumers, industry stakeholders, policymakers, and regulators about the complexities of various mobile contactless payment models and the underlying technologies. Education will help to dispel some of the myths about mobile payment security risks and demonstrate how mobile payments can be more secure than card or e-commerce payments.

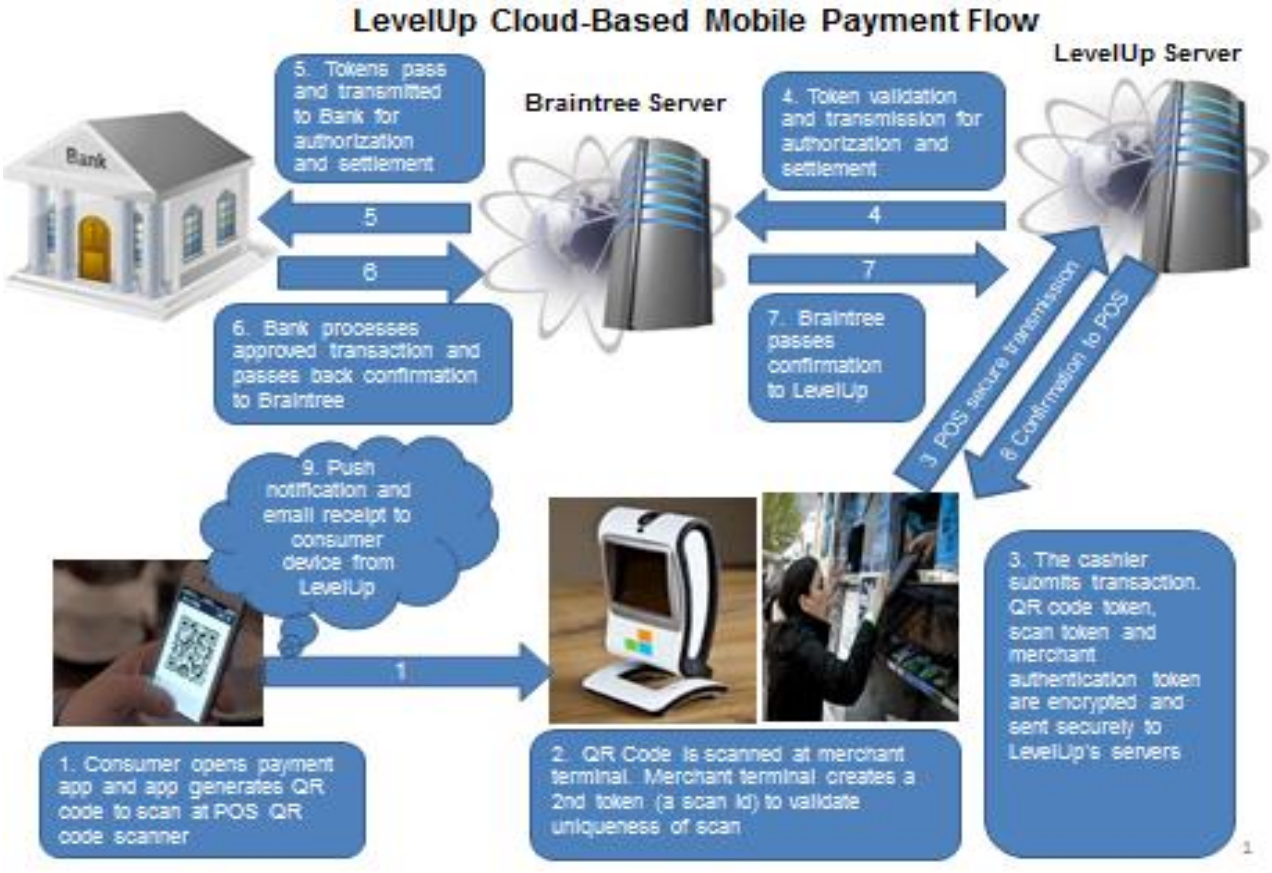
Appendix I: General NFC Payment Transaction Flow



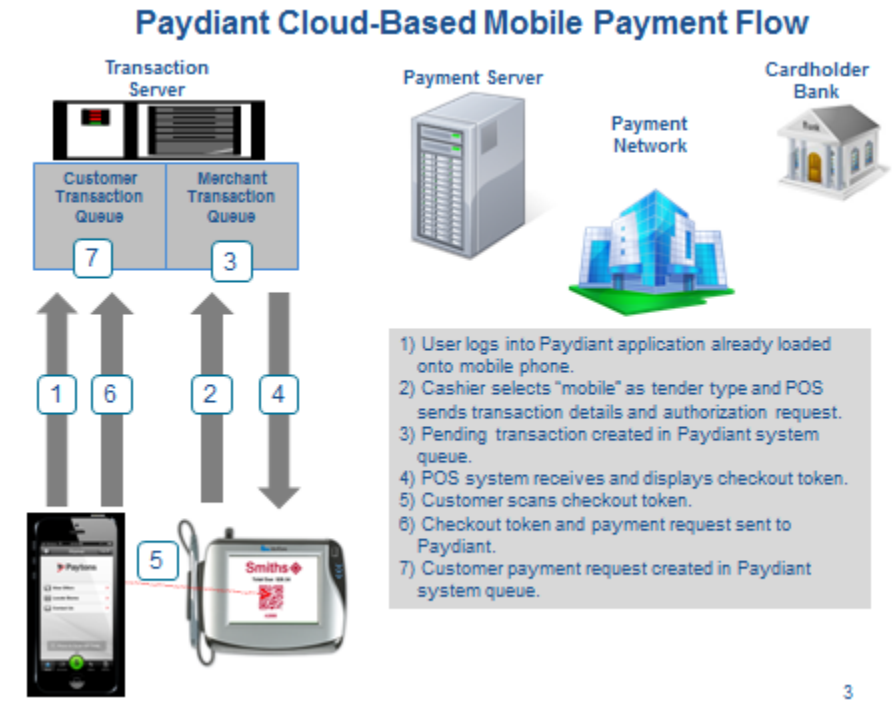
Appendix II: Comparison of NFC, QR Code, and Cloud Technology Platforms

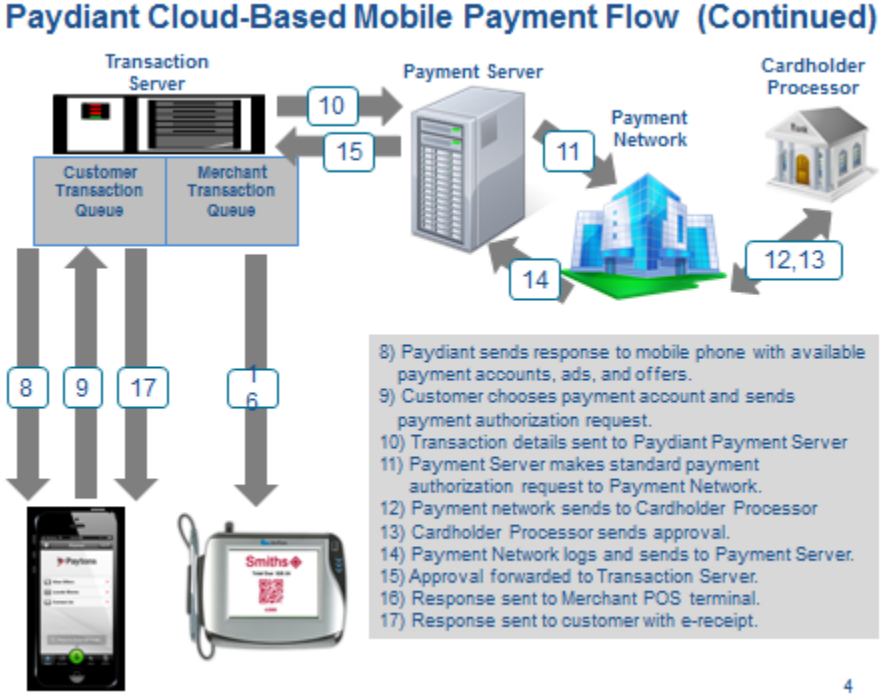
	NFC	QR Codes	Cloud
Issuance	• Secure Element to store payment credentials • TSM to manage provisioning	•Cloud-based mobile app	• Cloud-based mobile app • Payment credentials stored or accessed (tokenization) in cloud
Consumer Device Capabilities	• 9 of top 10 OEMs support NFC • 2-way wireless communication	•Only requires data connection •Not device dependent	• Only requires data connection • Not device dependent
Acceptance	• Standards based • Acceptance growing in developed countries • EMV may lead to further adoption	•Fragmented/many solutions •No standards •Security concerns •Requires fast wireless connection	• Fragmented • No standards • New customer experience • Security concerns • Requires fast wireless connection

Appendix III: LevelUp Cloud/QR Code Based Payment Model

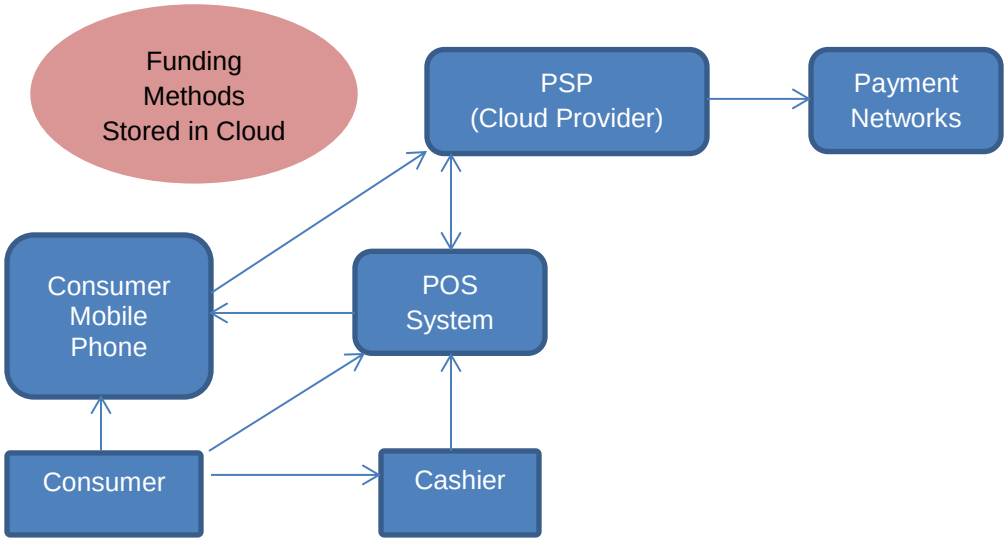


Appendix IV: Paydiant Cloud/QR Code Based Payment Transaction Flow

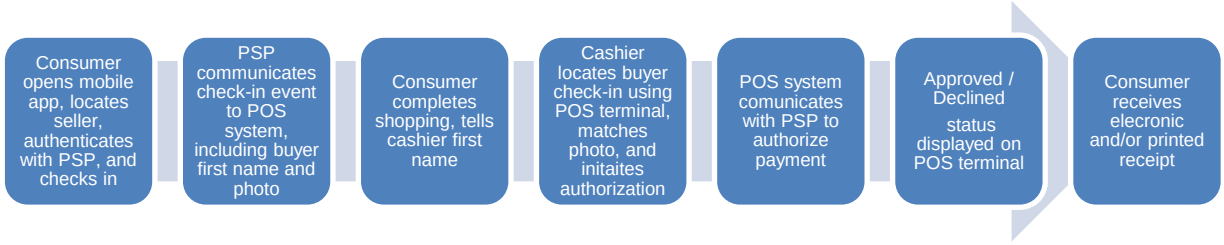




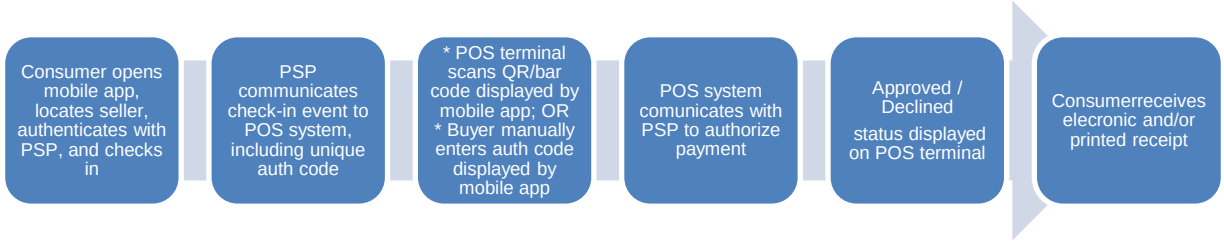
Appendix V: PayPal Cloud-Based Payment Transaction Flow



Name and Face:



Authorization Code:



Offline Payment Token: