FEDERAL RESERVE BANK OF BOSTON™

Payment Strategies

Understanding the Role of Host Card Emulation in Mobile Wallets
May 10, 2016

By Susan Pandy and Marianne Crowe, Federal Reserve Bank of Boston, Payment Strategies and Brian Russell, Giesecke & Devrient

The technology platforms that support the mobile payments landscape have evolved over the last few years. In the U.S., there are three models that leverage the near field communication (NFC)[1] protocol to support contactless mobile wallets.

(1) NFC with a secure element (SE)[2] embedded in the mobile phone that stores a payment token to replace the primary account number (PAN).
(2) NFC with host card emulation (HCE)[3] software that replaces the SE in the mobile phone to enable the NFC wallet app to perform card emulation. Payment tokens[4] are downloaded from a cloud server and stored in the mobile operating system (OS).
(3) NFC with a trusted execution environment (TEE), a secure area of the main processor in the mobile phone that stores the payment token.

This brief focuses on HCE mobile wallet models. Depending on which HCE model financial institutions (FIs) or other providers select, the payment credentials and associated cryptographic keys that generate the dynamic cryptogram for each transaction may be stored in the TEE or secure application memory of the mobile phone, not in the SE. However, storing payment credentials and cryptographic keys in the mobile phone OS instead of the SE is considered less secure, which is why additional security measures, including payment tokenization, are needed for HCE. The TEE, while not as secure as the SE, is a combination of hardware and software components, and considered more secure than the mobile OS.

HCE mobile wallet solutions can be implemented without tokenizing the PAN. However, in the U.S., NFC wallet solutions adhere to the *EMV Payment Tokenization Specification* (EMV spec)[5] for payment tokens, whether they use an SE or an HCE model.

[1] Near field communication (NFC) is a standards-based wireless communication technology that allows data to be exchanged between devices that are a few centimeters apart.

[2] 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 payment applications, the SE controls interactions between trusted sources (bank) and trusted applications (mobile payment app) stored on the SE and third parties (company the user is 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).

[3] The term "host card emulation" (HCE) was introduced in 2012 by SimplyTapp to describe the ability for a mobile wallet app to communicate through the NFC controller to a contactless NFC-enabled POS terminal/reader to pass payment card credentials (or payment token), eliminating the need for a physical SE managed by the mobile network operator (MNO). Research in Motion (RIM) had previously implemented a similar process on its Blackberry Bold 990 device in 2011, referring to it as "virtual target emulation."

[4] The term "payment tokens" refers to tokens as defined under the *EMV Payment Tokenization Specification* referenced in Footnote 5 below. Also, for more information about the tokenization and the difference between security (acquirer/processor) and payment tokenization (network/issuer), see Crowe, M., et. al. (2015, June). *Is tokenization ready for primetime? Perspectives from industry stakeholders on the tokenization landscape.* Available at http://www.bostonfed.org/bankinfo/payment-strategies/publications/2015/tokenization-prime-time.pdf.

[5] EMVCo (2014, March). *EMV payment tokenization specification – Technical framework.* Available at http://www.emvco.com/specifications.aspx?id=263.

The views expressed in this paper are those of the authors and do not necessarily represent those of the Federal Reserve Bank of Boston or the Federal Reserve System.
Mention or display of a trademark, proprietary product or firm does not constitute an endorsement or criticism by the Federal Reserve Bank of Boston or the Federal Reserve System and does not imply approval to the exclusion of other suitable products or firms.

Following the EMV spec, when a payment card is provisioned to a mobile wallet, the token service provider (TSP)[6] tokenizes the PAN and stores the token (whether on the SE, mobile OS, or TEE) in the phone. Additionally, MasterCard and Visa modified their contactless specifications to support single/limited use keys and cloud cryptograms that recognize HCE tokens as valid payment credentials.[7]

Of the three primary NFC models available in the U.S. today, Apple Pay stores payment tokens in the SE in the mobile phone (model 1);[8] Google Android Pay[9] uses HCE to store tokens in the Android KitKat v4.4 (or higher) mobile OS (model 2); and Samsung Pay uses NFC and HCE, but stores the payment token and cryptographic keys in the TEE in the mobile phone (model 3).[10]

The following section explains how HCE works, using Google Android Pay for context.

Android Pay uses HCE to replace the physical SE in the mobile phone with a virtual SE. Payment tokens and cryptographic keys are stored in the mobile OS, along with the mobile wallet app.

- To initiate an HCE mobile payment, a customer taps his mobile phone at the POS NFC reader.
- HCE enables the NFC controller in the mobile phone to route communications from the POS NFC reader to the mobile wallet app to request access to the payment token.
- The payment token and the dynamic cryptogram (generated by the cryptographic keys) are passed to the POS to complete the transaction.

A number of limited use tokens (i.e. session keys) pre-stored in the mobile OS enable the transaction to be completed without network connectivity. These keys are replenished each time the user is connected to a network. For added protection, limited use tokens are stored in an area of the mobile OS that uses software-based security, such as white box cryptography, to obfuscate a key. Storing it in the code of the cryptographic algorithm prevents exposure of confidential information.[11] Limited use tokens also have restrictions and expire quickly to minimize their value to fraudsters.

[6] The *EMV spec* defines a token service provider as "an entity that provides a token service comprised of the token vault and related processing." For now, only card networks can serve as TSPs but the specification is being updated to include requirements for non-network TSPs.

[7] Visa uses limited use keys derived from the master key and MasterCard uses single use keys (SUKs). Gartner (2015). *Samsung Pay will transform the mobile wallet experience.* Retrieved from http://www.samsung.com/hk_en/business-images/insights/2015/Samsung_Pay_Will_Transform_the_Mobile_Wallet_Experience-0.pdf. Multiple SUKs can be stored on a mobile device and as they are used additional SUKs are loaded from the cloud card management vendor to the device. Cryptomathic (2015). *Protect HCE mobile applications with cryptomathic MASC.* Retrieved from http://www.cryptomathic.com/hubfs/docs/protect_hce_mobile_apps_with_masc_v1.5.pdf?t=1457349447932.

[8] For more information on how Apple Pay works and the use of payment tokenization, see Crowe, M., et al. (2015, June). *Is tokenization ready for primetime? Perspectives from industry stakeholders on the tokenization landscape.* Available at http://www.bostonfed.org/bankinfo/payment-strategies/publications/2015/tokenization-prime-time.pdf.

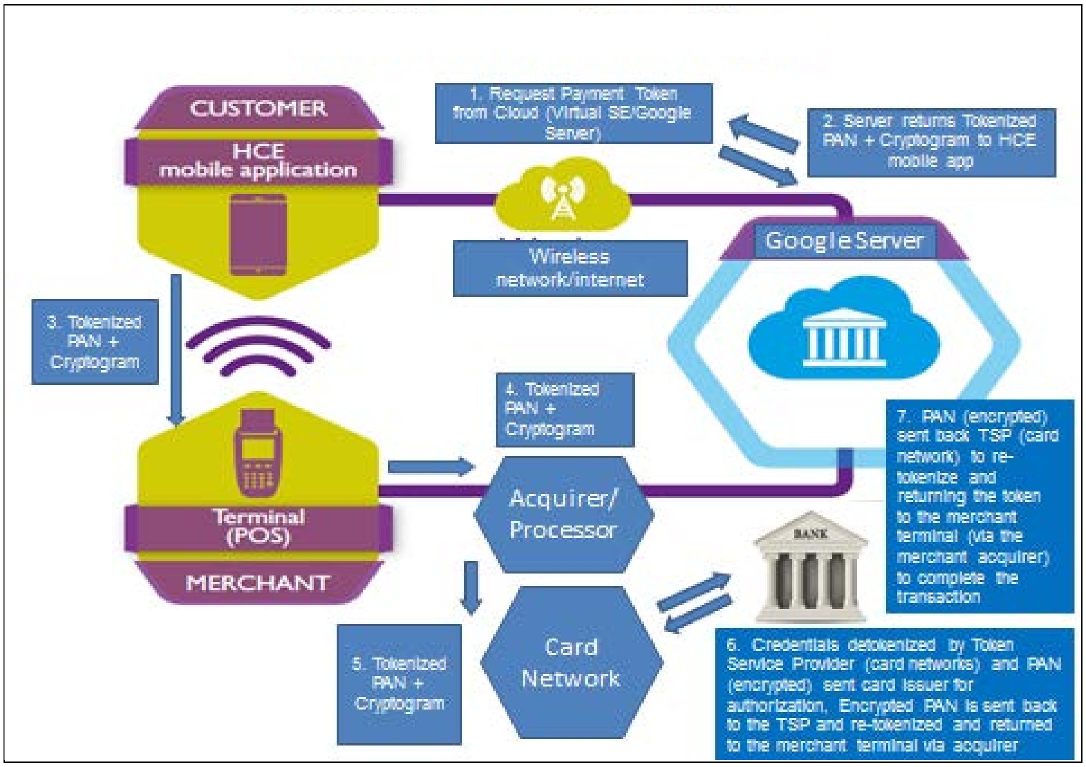
[9] Google Wallet was redesigned to support only person-to-person payments.

[10] Samsung Pay can also be used at non-NFC enabled POS terminals using its magnetic secure transmission (MST) technology, which emulates the magnetic stripe on a credit card and transmits this data using a magnetic field to the POS terminal.

[11] For more information about white box cryptography, see http://blog.bellid.com/what-is-white-box-cryptography.

Similar to Apple Pay, Android Pay follows the EMV spec. Figure 1 illustrates the payment token provisioning process and transaction flow using an NFC-enabled mobile phone with HCE.

Figure 1. HCE-based Payment Token Provisioning and Transaction Flow


CUSTOMER
HCE mobile application
1. Request Payment Token from Cloud (Virtual SE/Google Server)
2. Server returns Tokenized PAN + Cryptogram to HCE mobile app
Wireless network/internet
Google Server
3. Tokenized PAN + Cryptogram
4. Tokenized PAN + Cryptogram
7. PAN (encrypted) sent back TSP (card network) to re-tokenize and returning the token to the merchant terminal (via the merchant acquirer) to complete the transaction
Terminal (POS)
MERCHANT
Acquirer/ Processor
BANK
Card Network
5. Tokenized PAN + Cryptogram
6. Credentials detokenized by Token Service Provider (card networks) and PAN (encrypted) sent card issuer for authorization. Encrypted PAN is sent back to the TSP and re-tokenized and returned to the merchant terminal via acquirer

Source: FIME. (2015, August). *The NFC security quiz 2.0: Updated with HCE & tokenization*. Retrieved from https://www.fime.com/?/WhitePaper/view/7 as modified by the Federal Reserve Bank of Boston.

Provisioning

Step 1: Once the tokenized PAN (i.e., payment token) has been generated by the TSP, it is passed to the Google server to be provisioned to the HCE mobile app on the phone.

Step 2: The HCE mobile app requests the tokenized PAN and the Google server returns the tokenized PAN and cryptographic keys to the HCE mobile app.

Transaction Flow

Step 3: HCE mobile app transmits the tokenized PAN and cryptogram to the contactless POS terminal.

Step 4: Tokenized PAN and cryptogram are passed from the merchant POS to the acquirer/processor.

Step 5: Acquirer/processor passes the tokenized PAN and cryptogram to the card network.

Step 6: Card network/TSP de-tokenizes and sends the PAN over a secure/encrypted line to the issuer for authorization.

Step 7: Issuer sends the PAN back to the TSP to be re-tokenized and returned to the merchant POS terminal via the acquirer/processor to complete the transaction.

HCE Security Risks

In the HCE model, communication always passes through the Android OS. While there are basic security controls in this process they are limited, primarily because Android does not prevent rooting.[12] Rooting makes a mobile phone more susceptible to hacking and exposes HCE-based mobile payments to several vulnerabilities:[13]

(1) Rooting the mobile device can expose sensitive information such as payment credentials, and make it easier for malware to access the confidential data.
(2) Malware can root the device.
(3) Fraudsters can root a lost or stolen device to gain access to sensitive information stored in the wallet app and use it to conduct fraudulent payments.

SE-based NFC wallet apps will not work on rooted mobile phones, so these security risks are not present. However, there is an Android application for HCE wallets that tries to verify system settings and detect whether a device has been rooted. As a result, users who try to use Android Pay or Samsung Pay on a rooted phone will receive a message that the wallet app has been locked because of an unauthorized modification to the phone.[14] Given the risks associated with rooting, HCE wallet providers should make sure that their mobile wallet solutions check for this setting (developer options and root access) and take appropriate action as soon as the setting is detected.[15]

In addition to rooting, the primary risk associated with HCE is if the payment credentials, whether stored in the mobile OS, TEE, or in the cloud, are not tokenized. Cloud (i.e. software) environments are more vulnerable to security breaches and fraud than hardware (i.e., secure element) and must have stronger security controls to protect the PAN.

HCE also lacks standards. While Visa and MasterCard have released HCE specifications, no global standard exists to support interoperability, secure execution, remote management, and ubiquitous acceptance.

HCE Security/Risk Mitigations

How to prevent unauthorized access to the HCE mobile wallet application on the mobile device is a key challenge. This section discusses several tools to help to prevent unauthorized access.

Securing the storage location is one way to mitigate risk. Depending on the storage location the level of security may differ as described below. Storage locations for payment credentials in an HCE environment include: (1) host OS in the mobile device, (2) cloud-based SE, and (3) TEE.

Host: An application resides in the mobile Android OS to store and process payment credentials. This approach provides little security other than application sandboxing, which limits the environments in

[12] Rooting in the Android OS is comparable to jail-breaking in Apple iOS. Rooting allows mobile handset users to attain privileged control and access to the mobile device, customize the mobile OS, and circumvent restrictions applied by the device manufacturer or mobile carrier.

[13] UL. (2014, January). *HCE Security Implications, Analyzing the Security Aspects of HCE*. Retrieved from https://www.ul-ts.com/catalog/offerings/knowledge-sharing/white-papers-and-case-studies/landing/c-29/c-1684.

[14] Gokey, M. (2015, October 7). You'll need patience, but Verizon says Samsung Pay is coming to its phones. *Digital Trends*. Retrieved on October 16, 2015 from http://www.digitaltrends.com/mobile/samsung-pay-news/.

[15] *Ibid.*

which certain code can execute. Sandboxing isolates an application to prevent outside malware, intruders, system resources or other applications from interacting with the protected app.[16] Each app functions in its own sandbox and cannot access another app. The Android OS manages this process by assigning a unique user ID (UID) that allows only apps with the same UIDs to share resources. Host storage is considered the least secure option, and is not permitted by the Visa and MasterCard HCE specifications without the use of additional software security tools such as white box cryptography.

Cloud-based SE: This approach stores payment credentials as a master token in what is referred to as a "virtual SE" in the cloud, although some industry stakeholders consider this term inaccurate because a real SE is a physical chip. A cloud-based SE should use a hardware security module (HSM) in the cloud to store the master token used to virtualize the SEs. One could refer to the HSM as a "master SE" because HCE solutions do not have the benefit of an SE on the device and it is riskier to store the permanent (master) tokens in the mobile phone. Instead, a few tokens with limited use capability are derived from the master token and downloaded to the mobile phone and refreshed from the cloud after they are used. Storing a few limited use tokens in the phone, rather than requesting one from the cloud each time it is needed to make a payment, also addresses the possibility that an internet connection might not always be available to download a token. These limited use tokens (or keys) are stored in the mobile OS and generate cryptograms that are passed with an EMV payment token for each transaction.

TEE: This secure area of the main processor in the mobile phone ensures that sensitive data is stored, processed and protected in an isolated and trusted environment, using trusted security software. Isolating the TEE from the OS and its applications protects it from being compromised if the Android OS is rooted. While more secure than other options, the TEE does not have the level of security of an SE because it is not considered tamper-resistant. However, GlobalPlatform[17] specifications describe how applications can securely reside in the TEE, including how to interface trusted applications with the TEE, and how to communicate between applications running in the mobile OS and trusted applications residing in the TEE.[18]

In addition to secure storage, several other methods can enhance the security of HCE mobile payment transactions as outlined below:

User and hardware verification is performed by obtaining something the user knows (e.g., username/password or PIN), something the user has (e.g., device ID to validate the phone, smartcard reader, biometrics), and determining how the user behaves (e.g., multiple transactions made very quickly in several geographically distant locations could be denied).

Transaction constraints are similar to domain restrictions defined in the EMV spec for payment tokens. They can be used to limit transactions to certain channels (e.g., online v. POS), to specific merchants, or by dollar amount or country thus reducing token exposure and fraud risk.
Tokenizing or replacing the PAN with a substitute value increases the security of an HCE mobile payment, as noted earlier.

[16] Rouse, M. (2015). *Tech Target*.
[17] Global Platform is a certification authority. Its specifications are considered best practices endorsed by the industry and internationally recognized. For more information, see https://www.globalplatform.org/.
[18] Fime. (2015, August). *The NFC security quiz v2.0: Updated with HCE & tokenization*. Available for download at https://www.fime.com/?/WhitePaper/view/7.

Data analysis can provide real-time transaction assessments to monitor activity and identify anomalies. In effect, HCE must rely on third-party managed intelligence services that provide tools to strengthen authentication at the device and OS levels by leveraging big data ecosystems. The more data that can be used to measure and analyze, the better the overall security is.

*White box cryptograph*y prevents the key from being retrieved even if the original source code is available and could be used to hide payment credentials in the HCE application.

Complicating the security of HCE payments is that multiple entities (e.g., TSP, original equipment manufacturer, and wallet provider) are responsible for different HCE security layers.

Overall, while methods to secure HCE mobile payments exist, further testing and analysis needs to be done to ensure these payments are as safe as those using other methods.

HCE Mobile Wallet Implementations in Canada[19] and Australia

Royal Bank of Canada (RBC) is the largest card issuer in Canada, with 6.5 million cards. In January 2014, RBC launched its Secure Cloud platform, using an NFC SIM card in the mobile phone to store the payment applet and storing payment credentials in a private cloud. The Secure Cloud sent the payment request to the SIM payment applet and the NFC antenna. RBC added support for HCE to its mobile app in September 2015 to replace the SIM card model which was only supported by a few MNOs and mobile devices. Customers could load credit or debit cards to any mobile phones that supported Android KitKat 4.4 or higher, and pay any POS merchant that supported Interact Flash,[20] or Visa NFC contactless payments. Similar to RBC's earlier model, HCE mobile wallet payment credentials are stored securely behind firewalls in RBC's proprietary cloud, not on the phone. Use of the RBC mobile wallet among Canadians has been growing since implementation.

The *Commonwealth Bank of Australia* (CBA) incorporated a mobile payment service for its customers into its mobile banking app, using NFC and HCE on mobile phones running Android 4.4 KitKat or higher. CBA worked with Giesecke & Devrient on the integration. As of March 2015, CBA reported that its mobile banking app had 3.2 million registered users and had surpassed $100 billion in transactions.[21] To contain fraud, the app limits in-store purchases to $100 AUD.[22]

HCE appears to be gaining acceptance in other countries. TD Bank in Canada, Getin Bank in Poland, First Investment Bank AD in Bulgaria, ING in the Netherlands, and Banco Sabadell in Spain have integrated HCE into their mobile wallets.

[19] Crosman, P. (2015, April 10). Royal Bank of Canada forges its own path on mobile wallet, *American Banker*. Retrieved from http://www.americanbanker.com/news/bank-technology/royal-bank-of-canada-forges-its-own-path-on-mobile-wallet-1073704-1.html.

[20] Interac is a national payment network that allows Canadians to access their money through *Interac* Cash at ATMS, *Interac* Debit at POS terminals. *Interac* Flash is a secure contactless enhancement of *Interac* Debit that allows Canadians to pay for items instantly with their *Interac* chip debit card at a reader that supports *Interac* Flash. See https://www.interac.ca/en/interac-about/about-us.

[21] Clarke, S. (2015, March 11). Commbank turns on HCE, *NFC World*. Retrieved from http://www.nfcworld.com/2015/03/11/334558/commbank-turns-on-hce/.

[22] Ray. (2015, March 13). 19 banks/FIs that adopted HCE based NFC payments in the last six months. *Let's Talk Payments*. Retrieved from http://letstalkpayments.com/19-banksfis-that-adopted-hce-based-nfc-pay-in-last-six-months/.

While implementation of HCE is nascent, it allows issuing FIs to offer contactless mobile payment applications on the Android platform without needed business relationships with mobile network operators (MNOs) and the subsequent investment costs.[23] Visa and MasterCard support for HCE has helped to address some concerns, encouraging more FIs in the U.S. and globally to implement this model.[24] To date, Capital One Bank is the only U.S. bank to implement HCE.

Conclusion

HCE is still a nascent technology that has yet to match the level of security, standardization, and certification of SE-based NFC solutions. Card network payment tokenization services are addressing some of the security concerns that FIs in the U.S. and other countries face when implementing this wallet model. However, FIs implementing HCE-based wallets need to make sure their plans also include other security measures, such as those outlined in this paper.

HCE also affords stakeholders a faster, more flexible and cost-effective solution. HCE eliminates the need for an MNO-controlled SIM/SE and trusted service manager (TSM)[25] to manage SE provisioning. And, once HCE is implemented, FI customers can immediately download the wallet app and begin using it – assuming they have the minimum OS requirements and NFC hardware.

The current mobile payments market supports both NFC/SE-based and HCE/cloud-based mobile payment platforms. It is too soon to predict if HCE will prove to be a viable alternative to the SE/NFC model and whether one model will prevail, or if both can co-exist to address different payment needs. Fortunately, there are now commercially available U.S. implementations of HCE with Android Pay and Samsung Pay that industry stakeholders can monitor and analyze for performance and security.

[23] Thales e-Security (2014, October). *Creating trust infrastructure for mobile payments*. Retrieved form http://images.go.thales-esecurity.com/Web/ThalesEsecurity/%7B7d827247-4f39-4ecf-9686-c39455f20e18%7D_Creating_Trust_Infrastructure_for_Mobile_Payments_wp.pdf

[24] Hernandez, W. (2015, April 17). The rise of HCE-based mobile payments. *Mobilepaymentstoday*. Retrieved from http://www.mobilepaymentstoday.com/articles/this-rise-of-hce-based-mobile-payments/.

[25] A trusted service manager (TSM) acts as a neutral broker in the NFC ecosystem by establishing business agreements and technical connections with MNOs, phone manufacturers or other entities controlling the SE on mobile phones. The TSM enables service providers to distribute and manage their contactless payment applications remotely by allowing access to the SE in NFC-enabled mobile phones.