WHITE PAPER
Smart Card Alliance
Mobile & NFC Council

A SMART CARD ALLIANCE MOBILE & NFC COUNCIL WHITE PAPER

Host Card Emulation (HCE) 101

Publication Date: August 2014
Publication Number: MNFCC-14002

Smart Card Alliance

191 Clarksville Rd.
Princeton Junction, NJ 08550

www.smartcardalliance.org

About the Smart Card Alliance

The Smart Card Alliance is a not-for-profit, multi-industry association working to stimulate the understanding, adoption, use and widespread application of smart card technology. Through specific projects such as education programs, market research, advocacy, industry relations and open forums, the Alliance keeps its members connected to industry leaders and innovative thought. The Alliance is the single industry voice for smart cards, leading industry discussion on the impact and value of smart cards in the U.S. and Latin America. For more information please visit http://www.smartcardalliance.org.

The Smart Card Alliance has used best efforts to ensure, but cannot guarantee, that the information described in this report is accurate as of the publication date. The Smart Card Alliance disclaims all warranties as to the accuracy, completeness or adequacy of information in this report. This white paper does not endorse any specific product or service. Product or service references are provided to illustrate the points being made.

Smart Card Alliance

Table of Contents

1 Introduction

Google's adoption of host card emulation (HCE) in the Android operating system (OS) v4.4 (KitKat) has created a new market opportunity for solution providers and issuers to implement and deploy NFC solutions, removing both dependencies on the secure element (SE) and trusted service manager (TSM) infrastructure and the need to set commercial agreements with secure element issuers (SEIs). HCE support is currently available in the Android operating system (Android KitKat 4.4 and higher) and the BlackBerry operating system.

While the availability of HCE creates the opportunity for anyone to develop a mobile device application that can function as a smart card, it also introduces security considerations.

This white paper was developed by the Smart Card Alliance Mobile and NFC Council to provide an educational resource on HCE. The white paper presents NFC/HCE technology, describes how it is used, outlines key considerations for payments use cases, discusses the security considerations associated with HCE implementation, outlines various HCE use cases for both payment and non-payment applications, and compares HCE and SE-enabled implementations.

2 Near Field Communication and Host Card Emulation

Near Field Communication (NFC) is a short-range wireless (RF) communication technology for smartphones and similar devices that enables data transfer between the devices. NFC operates at 13.56 MHZ, complies with the ISO/IEC 14443 and ISO/IEC 18092 standards, and MIFARE and FeliCa specifications, and operates in ranges of less than 10 cm.[1]

NFC in conjunction with a mobile wallet or a use-case-specific application (app) is used for a variety of applications, such as payment, ticketing, access, RFID tags, loyalty, and coupons, as well as in consumer electronics. Currently, NFC-based applications that use the card emulation mode (i.e., where the reading terminal effectively sees the mobile phone mimicking a traditional contactless smart card and no change to the reading infrastructure is required) require the card application (e.g., payments, ticketing, access control) and its credentials (e.g., account information, ticket, access identifier and tokens) to be stored inside a hardware-based secure element (SE[2]) on the mobile device.

Host card emulation (HCE) enables NFC devices to perform contactless transactions in card emulation mode when the payment, other credentials and related card applications are stored somewhere other than the SE: e.g., in the cloud, in a trusted execution environment on the mobile device, or in a virtual, software-based infrastructure on the mobile device. NFC was defined by the NFC Forum and is included in NFC Forum specifications, including the NFC controller interface (NCI) specification, which in combination with standards such as ISO/IEC 14443 and JIS X 6319-4, enable HCE implementation.[3]

2.1 Current NFC Support

The availability of NFC-enabled mobile devices continues to grow, with NFC included in over 345 million devices shipping in 2013.[4] According to NFC World,[5] over 50 manufacturers support NFC in over 200 phone models and tablets. Manufacturers currently include Acer, Asus, BlackBerry, Casio, Fujitsu, Google, HP, HTC, Huawei, Lenovo, LG, Motorola, Nokia, Panasonic, Pantech, Samsung, Sharp, and Sony. Although Apple mobile devices do not currently support NFC, NFC capability can be added to iPhones through commercially available cases and accessories.

Numerous NFC-enabled payment, marketing, ticketing, and other applications have been implemented globally.[6] In the United States, the most prominent NFC mobile payment services are offered by Isis and Google.

Isis[7], the mobile carrier joint venture that includes AT&T, Verizon, and T-Mobile, offers the Isis Wallet, which supports mobile payments, loyalty programs, and offers. A total of 68 mobile phones support NFC and the Isis

[1] Additional information on NFC can be found on the NFC Forum Web site, http://www.nfc-forum.org.

[2] The SE is a tamper-resistant smart card module that can store data and applications and execute applications securely. The SE can be used in multiple application environments and can be available in multiple form factors. The SE includes a microcontroller CPU, an operating system, different types of memory (ROM, EEPROM, and RAM), and cryptographic engines.

[3] "NFC Forum Statement Regarding Host Card Emulation (HCE)," March 20, 2014, http://nfc-forum.org/newsroom/nfc-forum-statement-regarding-host-card-emulation-hce/#_ftn2.

[4] Karl Dyer, "ABI: Smartphones accounted for 80% of the NFC devices shipped in 2013," *NFCWorld+*, Jan. 8, 2014, http://www.nfcworld.com/2014/01/08/327447/abi-smartphones-accounted-80-nfc-devices-shipped-2013/.

[5] "NFC phones: The definitive list," *NFCWorld+*, May 11, 2014, http://www.nfcworld.com/nfc-phones-list/.

[6] Additional information can be found at http://www.nfcworld.com/list-of-nfc-trials-pilots-tests-and-commercial-services-around-the-world/.

[7] Note that Isis has announced that they will be re-branding. "A Message from Michael Abbott: Embarking on a New Brand," July, 7, 2014, http://news.paywithisis.com/2014/07/07/isis-wallet-rebranding/.

Wallet[8]; consumers with an iPhone must use an Isis Ready case.[9] Consumers set up an American Express Serve account or add participating American Express, Chase, or Wells Fargo cards to the Isis Wallet. The Isis Wallet can then be used to pay at any merchant accepting contactless payment. Isis stores the payment applications and account information in the SE that is built into the mobile phone's hardware.

Google, in partnership with Sprint, introduced the Google Wallet in 2011. Google Wallet supports payment, loyalty, money transfer, offers, and online order tracking. In the latest version of the Google Wallet, the payment functionality requires an NFC-enabled device running Android 4.4 (KitKat) or higher on any carrier network.[10] The initial Google Wallet implementation relied on an SE in the mobile phone's hardware; Google has since changed directions, using HCE with payment credentials stored in the cloud. When the consumer taps the phone, HCE enables Google Wallet to pass transaction information to the point-of-sale (POS) terminal to complete the transaction.[11]

2.2 NFC Operating Modes

The NFC Forum technical specifications define three NFC operating modes: reader/writer, peer-to-peer, and card emulation.

Reader/writer mode enables NFC devices to read and write information to NFC tags (e.g., in posters or advertisements). In this mode, NFC devices can read NFC Forum-mandated tag types, which are compliant with the NFC-A, NFC-B, and NFC-F specifications.

Peer-to-peer mode enables NFC devices to exchange data and share files. Peer-to-peer mode complies with ISO/IEC 18092 and may use the NFC Forum's Logical Link Control Protocol specifications to enable bidirectional data transfer.

Card emulation mode (which is the mode that is the focus of this white paper) enables NFC devices to function as contactless smart cards complying with the ISO/IEC 14443 standard and FeliCa specification. Consumers can conduct transactions such as purchasing, ticketing, and accessing transit with a tap of the device. NFC-enabled devices complete contactless transactions using the current contactless acceptance infrastructures.

Figure 1 summarizes the three NFC operating modes and their related standards.

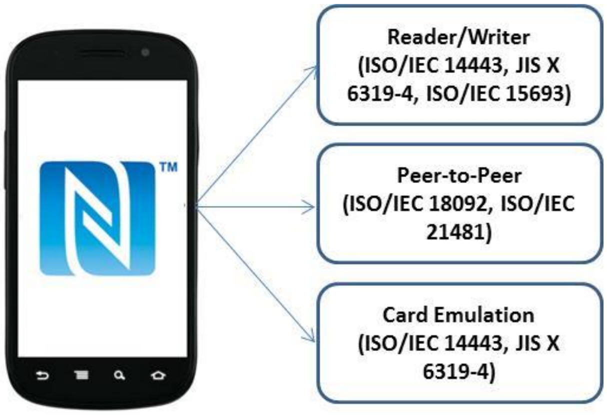
TM
Reader/Writer
(ISO/IEC 14443, JIS X 6319-4, ISO/IEC 15693)
Peer-to-Peer
(ISO/IEC 18092, ISO/IEC 21481)
Card Emulation
(ISO/IEC 14443, JIS X 6319-4)

Figure 1. NFC Operating Modes

[8] "Isis Reports 600K New mWallet Downloads In The Last Month," May 14, 2014, PYMNTS.com, http://www.pymnts.com/news/2014/isis-reports-a-600k-new-mwallet-downloads-in-the-last-month/?utm_source=May+15%2C+2014&utm_campaign=US+NL+May+15%2C+2014&utm_medium=email

[9] http://www.paywithisis.com.

[10] https://support.google.com/wallet/answer/1347934?hl=en.

[11] http://www.google.com/wallet/faq.html#tab=faq-security.

2.3 Card Emulation Mode

In card emulation (CE) mode, a mobile device can emulate any contactless smart card[12] (such as those used for contactless payments, transit fare payment and building or hotel room access) when tapped on a contactless reader or point-of-sale (POS) terminal.

Until recently, the virtualized contactless card application and associated credentials were always stored in a secure element (SE), defined by GlobalPlatform as: “a tamper-resistant platform (typically a 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 a set of well-identified trusted authorities.”

HCE opens up the possibility of storing the virtual contactless card application as a service running on the mobile device operating system. This option is supported today by Google’s Android OS (v4.4 “KitKat” onwards) and by the Blackberry OS.

2.3.1 SE-Enabled Card Emulation

With SE-enabled card emulation, the NFC controller routes the communication from the contactless reader or POS terminal to a tamper-resistant dedicated hardware component called the SE. The SE safely stores the card emulation application and associated credentials.

All NFC-enabled mobile devices implement the capability to allow SEs to:

1. Communicate with the NFC controller, and through it, with contactless readers to perform transactions.
2. Communicate with user-interfacing mobile applications running on the mobile device operating system, such as mobile wallets.
3. Communicate over-the-air with the credential provisioning infrastructure, called the trusted service manager (TSM).

Figure 2 illustrates the mobile device architecture for SE-based NFC card emulation.

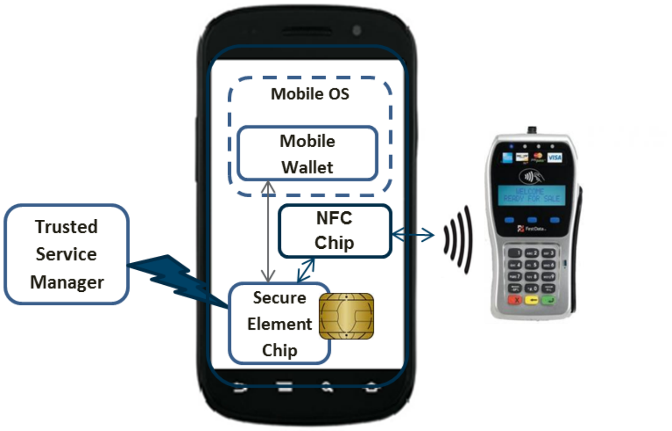

Figure 2. SE-Based NFC Card Emulation

[12] Footnote: Smart card technology is available in multiple form factors, including plastic cards, fobs, and secure elements used in mobile phones. This white paper uses the term “smart card” generically to refer to all forms of smart card technology.

The secure element can reside in an embedded secure smart card chip on the handset, on the Subscriber Identity Module (SIM) or Universal Integrated Circuit Card (UICC), or on a secure digital (SD) card that can be inserted into the mobile phone. SIMs and UICCs are issued by the mobile network operators (MNOs) and embedded SEs are issued by mobile device manufacturers. SEs on microSD cards can be issued by any application provider.

When credentials are stored in the SE, they are provisioned by an entity known as a TSM. Provisioning the NFC SE application and related credentials requires cooperation and integration among multiple entities, which may include issuers, wallet providers, MNOs, payment processors, TSMs, and other members of the ecosystem.

Figure 3 illustrates the use of a TSM to provision SE applications and credentials.

Credentials stored in an SE are stored in security domains that adhere to GlobalPlatform specifications. Each service provider or issuer is assigned a specific domain, and each domain is protected by cryptographic keys that are known only to the participants, protecting them from any unauthorized access. During a payment transaction, the mobile wallet application authenticates itself to the SE, typically through a PIN or password, key, or digital signature, to enable transmission of the credentials to a contactless POS terminal or other acceptance device.

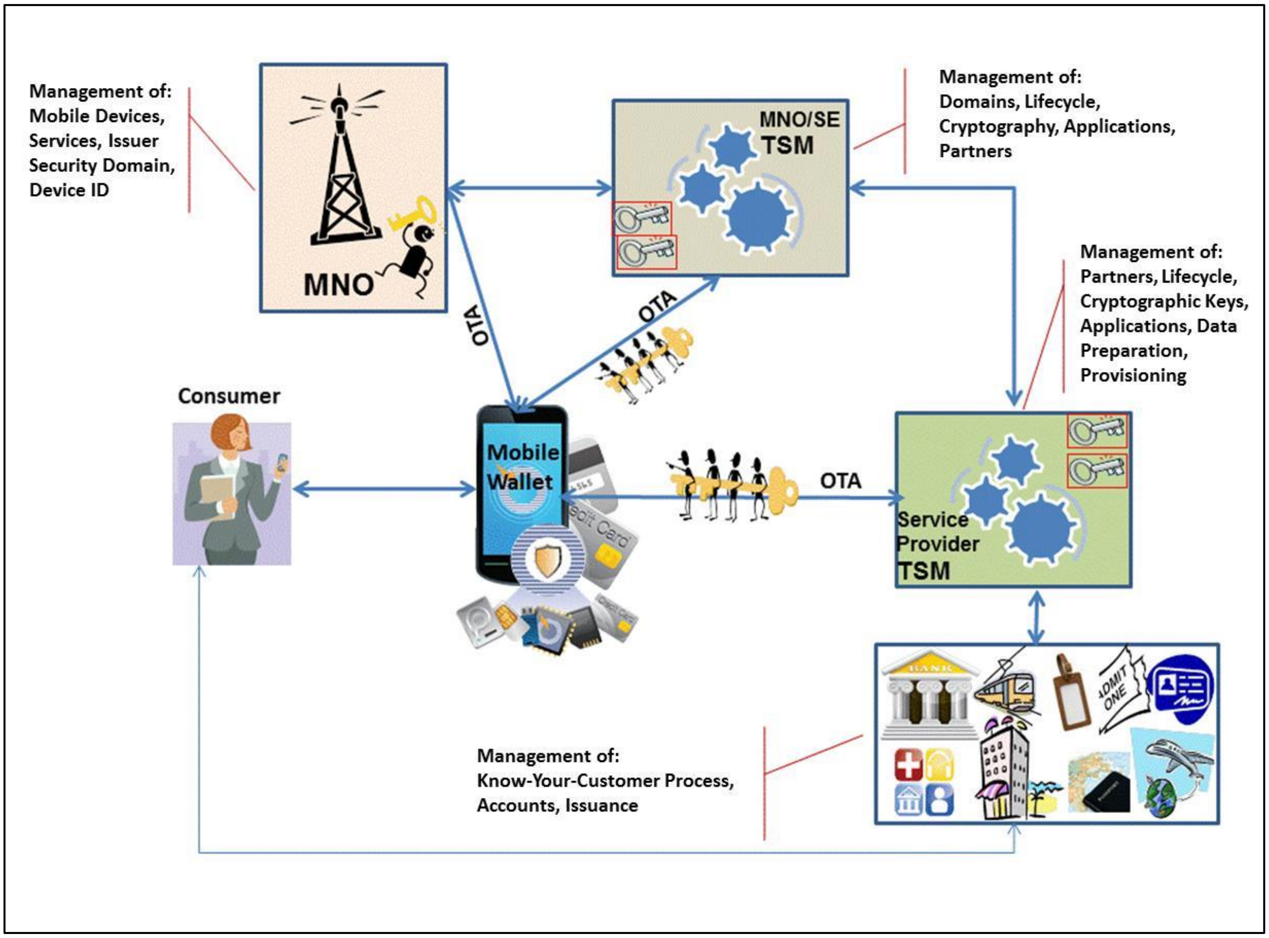

Figure 3. Use of a TSM to Provision Credentials to the SE

2.3.2 Host Card Emulation

HCE introduces an option for the NFC controller to now additionally route communication from the contactless reader or POS terminal to an HCE service on the mobile device's host CPU. With HCE, an 'APDU Service' running on the host can interface with a contactless reader via NFC. This HCE service can be part of a mobile application with a user interface, such as a mobile wallet for payment. The credentials used by this HCE service can be stored in the application itself, or they could be stored in other secure locations such as a trusted execution environment (TEE) or an SE (see Figure 4).

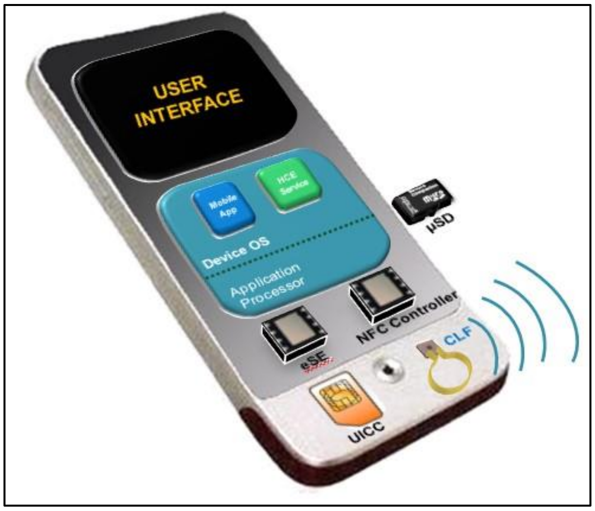

Figure 4. HCE Service with Different SE Form Factors

Alternatively, the HCE service could connect in real-time or at given intervals with a back-end server in the cloud to retrieve credentials to exchange with the contactless terminal. Real-time retrieval of credentials from the cloud at the moment of tapping on a reader is a possible but unlikely option, as network latency may result in a poor user experience.

Figure 5 illustrates this process for a payment app.

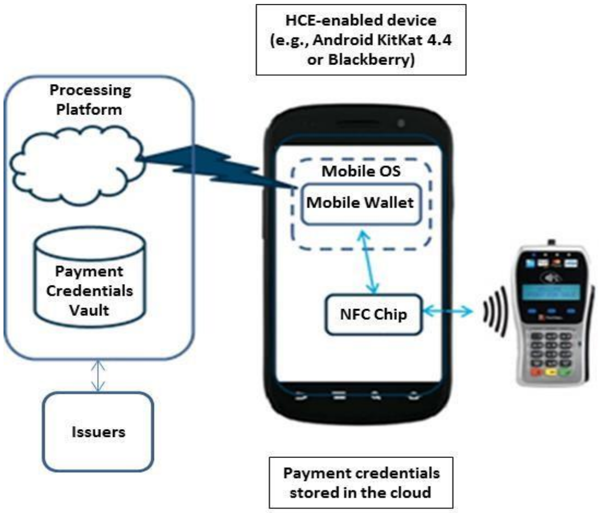
HCE-enabled device (e.g., Android KitKat 4.4 or Blackberry)
Processing Platform
Payment Credentials Vault
Issuers
Mobile OS
Mobile Wallet
NFC Chip
Payment credentials stored in the cloud

Figure 5. Obtaining Credentials from the Cloud Using HCE

2.3.3 Application Identifier Routing

Prior to the introduction of HCE, all requests coming from a contactless reader to communicate with an NFC application were routed to the active SE. The HCE implementation in Android or other mobile operating systems must take into account the possible coexistence of NFC card emulation services on an SE and in the host (mobile device) OS. To do so, Android KitKat defines a procedure called 'AID routing' that will allow the NFC controller to determine where to route a request from a reader to communicate with a given NFC application, which is identified by its application identifier (AID). The NFC controller implements a routing table, populated by the mobile OS, which lists the AIDs of NFC applications stored in the SE.

When the NFC controller receives a request to select an application ('SELECT AID' command) from a contactless reader, it searches for the AID in its routing table. If it finds it, it will route the command to the active SE; otherwise, it will take it to the host.

Figure 6 summarizes the two approaches to providing credentials for a transaction. On a mobile device that does not support HCE, all calls coming from a contactless reader are routed to the active SE. On a mobile device with HCE support, the NFC controller acts as a switch and routes the information appropriately to either the SE or the host OS.

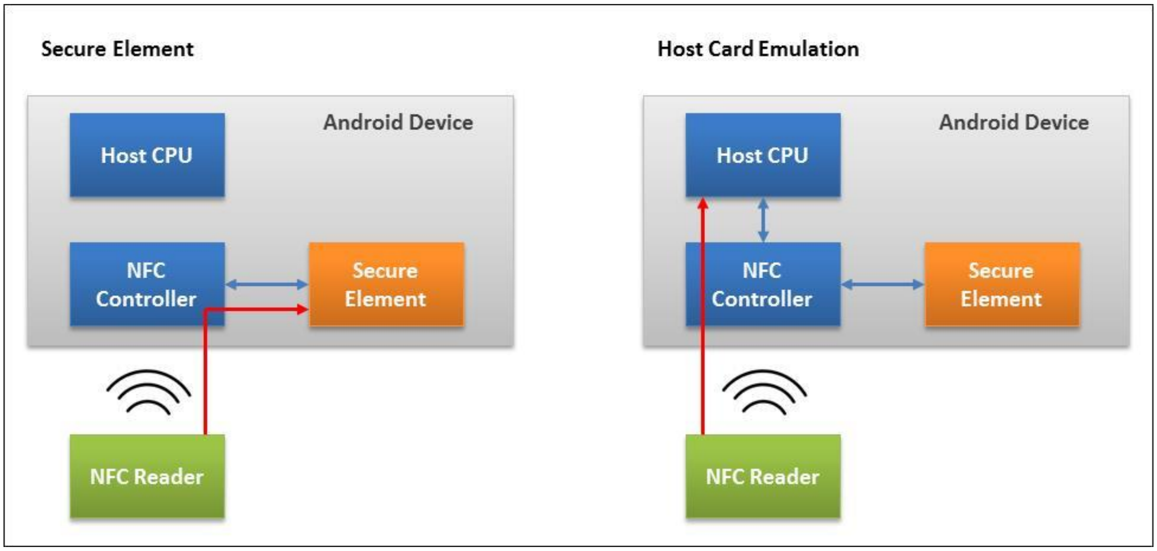

Figure 6. NFC Communications with the SE (left) or with the Host CPU Using HCE (right)

A summary of areas of consideration for SE and HCE implementations, focusing on areas where there are differences between the approaches, can be found in Appendix A.

3 Considerations for Payment Applications

The payments industry has been anticipating the use of NFC for mobile payments at the physical POS since the technology was introduced. Commercial implementations are now available in the United States and internationally.[13]

An NFC-enabled mobile payment transaction typically follows these steps:

1. An NFC-enabled mobile phone is provisioned with a wallet, which is used to interact with: a payment application for an SE implementation (e.g., American Express® ExpressPay, Discover Zip, MasterCard® PayPass, Visa payWave); or a payment account identifier for HCE; and a payment account (credit, debit, or prepaid) issued by the consumer's financial institution (the 'emulated card').

2. To complete a purchase, the consumer opens the wallet, selects the payment account to use, and holds or taps the phone close to the merchant's card reader. The consumer may be asked to enter a personal identification number (PIN) to enable the mobile payment app.

3. The transaction between the POS reader and the emulated card takes place exactly as with a physical contactless payment card. With an SE implementation, the authorization and settlement processes that follow are the same processes used when a consumer pays with a traditional contactless, EMV, or magnetic stripe credit or debit card. With an HCE implementation that uses tokens (see Section 3.2.2), an additional step may be required to associate the token with the original payment account.

To date, implementations of NFC-enabled mobile payments have relied on the SE in the mobile phone to facilitate transaction authentication and security, and to provide secure memory to store payment applications and account information. Provisioning the payment application and payment account to the mobile phone is done through an infrastructure that includes the account issuer, one or more TSMs, and MNO.

3.1 Current Support for HCE-Enabled Payments

This section discusses a few commercial implementations of HCE-enabled payments have been announced recently.

- Google Wallet in the United States
- Tim Hortons in Canada
- MasterCard and Visa support for payment apps using HCE that comply with their contactless payment specifications
- BBVA in Spain

3.1.1 Google Wallet

Google Wallet for NFC payments was initially implemented using an SE-based model. Shortly after the release of the new Android KitKat OS, Google Wallet appeared using an HCE implementation in November 2013 on the first KitKat device to reach the market, the Nexus 5. Since then, Google has switched to an HCE implementation and on April 14, 2014, stopped supporting the SE-based version.[14]

[13] Additional information on international NFC applications can be found at http://www.nfcworld.com/list-of-nfc-trials-pilots-tests-and-commercial-services-around-the-world/.

[14] Sarah Clark, "Google Wallet ends support for physical secure elements," *NFC World*$^{+}$, Mar. 17, 2014, http://www.nfcworld.com/2014/03/17/328326/google-wallet-ends-support-physical-secure-elements/.

Google Wallet's tap-and-pay feature is available at U.S. retail locations supporting contactless payments and requires an NFC-enabled device running Android 4.4 (KitKat) or higher on any carrier network.[15]

Google has confirmed its move to HCE: "Host card emulation allows Android applications to communicate directly over NFC on supported devices with Android 4.4 KitKat. When you tap your phone to pay, HCE enables Google Wallet to pass transaction information to the point-of-sale terminal to complete your transaction."[16] Devices that are running older operating systems may no longer support Google Wallet's tap-and-pay feature.

The user experience is similar in both the SE and HCE implementations. Google also states that payment credentials are not shared with merchants and are stored securely on a remote server.[17] Because Google stores payment account information in the cloud and actual card numbers are not shared with retailers, this approach could be viewed as a form of tokenization, as opposed to storing actual credentials on the mobile device, minimizing the chance of compromising credentials at a merchant location. (Additional information on tokenization can be found in Section 3.2.2.) This service also provides a purchase history (merchant details, transaction time and amount, optional geolocation information) that can be reviewed using the wallet app.

3.1.2 Tim Hortons

Tim Hortons[18] mobile payments service enables customers to tap to pay with a BlackBerry 10 smartphone, using a mobile version of the chain's closed-loop Tim Card (similar to a Starbucks card). Because Apple devices do not natively support NFC, QR codes are used to accommodate customers using iOS devices (running iOS version 6.0 and higher).

3.1.3 Visa and MasterCard

In February 2014, Visa announced it is "offering clients new options to securely deploy mobile payment programs, including for the first time an option to host Visa payWave-enabled accounts in a secure, virtual cloud," leveraging the HCE architecture. As part of this support, Visa is introducing new standards, tools, and guidelines for payWave HCE support.[19] The initial Visa payWave standard for cloud-based deployment is available.

MasterCard also announced that it is publishing specifications that leverage HCE for secure NFC payment transactions. MasterCard's press release indicated that HCE pilot projects are already underway with Capital One in the U.S. and Banco Sabadell in Europe.[20]

The MasterCard and Visa implementations use EMV-compliant messaging to ensure compatibility with installed EMV contactless POS terminals.

3.1.4 BBVA

BBVA announced that they have commercially launched a host card emulation-based mobile contactless payments service in Spain taking advantage of Visa's support for HCE.[21] BBVA has updated its Android wallet app for Spanish customers and is planning to introduce the technology in the U.S., Mexico and Chile.

[15] https://support.google.com/wallet/answer/1347934?hl=en.

[16] Google Wallet FAQ, http://www.google.com/wallet/faq.html#tab=faq-security.

[17] Google Wallet FAQ, http://www.google.com/wallet/faq.html#tab=faq-security.

[18] "Tim Hortons launches NFC payments service using Host Card Emulation," NFC World, December 13, 2013, http://www.nfcworld.com/2013/12/13/327339/tim-hortons-launches-nfc-payments-service-using-host-card-emulation/.

[19] "Visa to Enable Secure, Cloud-Based Mobile Payments," Visa press release, February 19, 2014, http://investor.visa.com/news/news-details/2014/Visa-to-Enable-Secure-Cloud-Based-Mobile-Payments/default.aspx.

[20] "MasterCard to Use Host Card Emulation (HCE) for NFC-Based Mobile Payments," MasterCard press release, Feb. 19, 2014, http://newsroom.mastercard.com/press-releases/mastercard-to-use-host-card-emulation-hce-for-nfc-based-mobile-payments/.

[21] "BBVA introduces HCE-based mobile payments," Finextra, June 30, 2014, http://www.finextra.com/news/fullstory.aspx?NewsItemID=26217.

3.2 Credential Storage and Security Using HCE

HCE does not rely on an SE for credential storage. To enable HCE, the payment credentials can be stored on the device as a static image, or they can be dynamically updated by a secure server in the cloud, based on the business rules associated with the payment application. However, serious security issues are associated with storing static data on the device.

There are at least two approaches to protecting sensitive payment card credentials stored on a cloud-based system during transactions: transactions without tokenization[22] and transactions with tokenization. In both approaches, all of the card information is stored securely in the cloud. The cloud-based server can be hosted by a wallet provider, a retailer, or a bank (or a processor on behalf of a bank or a retailer).

3.2.1 Cloud Storage without Tokenization

An HCE-enabled NFC payment transaction without tokenization proceeds as follows:

1. The customer registers and acquires the card credentials either through a mobile app or using the provider's secure Web-based service (such as a bank's Internet banking site).
2. At time of payment, the customer is authenticated by the cloud (using the credentials entered on the mobile app). The customer then selects a card to use for payment from the mobile wallet app.
3. The payment credentials are sent to the customer's mobile device to initiate the transaction. The device transmits the payment credentials to the merchant using NFC.

This solution is not considered secure. The payment credentials can be exposed by malware resident on the device. This approach is unlikely to find favor with the global payment brands.

3.2.2 Cloud Storage with Tokenization

An HCE-enabled NFC payment transaction with tokenization would proceed as follows:

1. The issuer offering the mobile payment service guides the customer through an enrollment/registration process that incorporates strong authentication methods and the consumer downloads the mobile payment app.
2. The customer's mobile app is provisioned with payment tokens, which may have limited validity depending on the business rules defined by the issuer. For example, the token may be:
   - Valid only for transactions not exceeding a certain amount threshold
   - Limited to a single use (after which a new token must be provisioned to the mobile device)
   - Valid only for a limited time
   - Valid only for mobile contactless payment transactions
   - Valid only at a given merchant or merchants

   Provisioning of the payment tokens is typically carried out in advance of any payment transactions to avoid increasing transaction latency times. Moreover, provisioning the payment tokens requires the mobile device to have data connectivity, which may not be available to the customer when the payment transaction is initiated.

[22] Tokenization in payments replaces the primary account number (PAN) with a surrogate value that is used in transactions in place of the PAN.

3. At time of payment, the customer's mobile payment app provides the tokenized payment credentials to the merchant's POS using NFC. The customer may be prompted to enter a PIN specific to the use of the mobile payment app.
4. The merchant routes the transaction to the acquirer, and the transaction is ultimately received by the issuer (over the payment network) for authorization. The issuer (or an entity acting as the token vault on behalf of the issuer) authorizes the transaction after verifying the token and identifying the associated payment credentials.

Note that this solution would also need to consider how to securely provision the payment token to the device.

This solution does not reduce the risk of credential exposure due to malware on the device, but reduces the impact of eventual exposure by replacing the static payment credential with a token of much reduced scope. Malware risk is increased if the device is exploited, jailbroken or rooted.[23]

To enhance security, implementations may use white box cryptography and software tamper proofing to replace SE functionality by software and be supplemented by techniques such as device fingerprinting and risk-scoring mobile payment transactions prior to authorization. Additional information on security approaches is included in Section 4.

3.3 Routing Mechanism for Multiple Payment Implementations in a Mobile Device

HCE has the potential to increase use of NFC. However, some issuers may still prefer the SE-based payment approach. For this reason, while HCE simplifies NFC implementation by eliminating the requirement for an SE, smartphones supporting HCE will still have to support SE-based transactions. This will require the smartphone to be able to select the appropriate application.

When a customer chooses a payment app that stores credentials inside the SE, the NFC controller routes the transaction to the SE. In HCE-based transactions, the communication is routed to the host processor in the mobile phone.

The Android KitKat 4.4 HCE service routes transactions using application and service selection methods. Application coexistence is enabled by means of AID-specific routing. The NFC controller on the device maintains a routing table that lists AIDs for some of the applications on the device and their associated routing rules. When a contactless reader sends an ISO 'select AID' command to the device, the NFC controller identifies the particular application and matches it to a rule in the routing table. Based on that rule, the destination is then identified as host processor (using HCE) or SE, and commands are routed accordingly (Figure 7).

[23] Rooting allows someone using a device running the Android operating system to attain privileged control. Jailbreaking is the process of removing limitations on iOS, Apple's operating system.

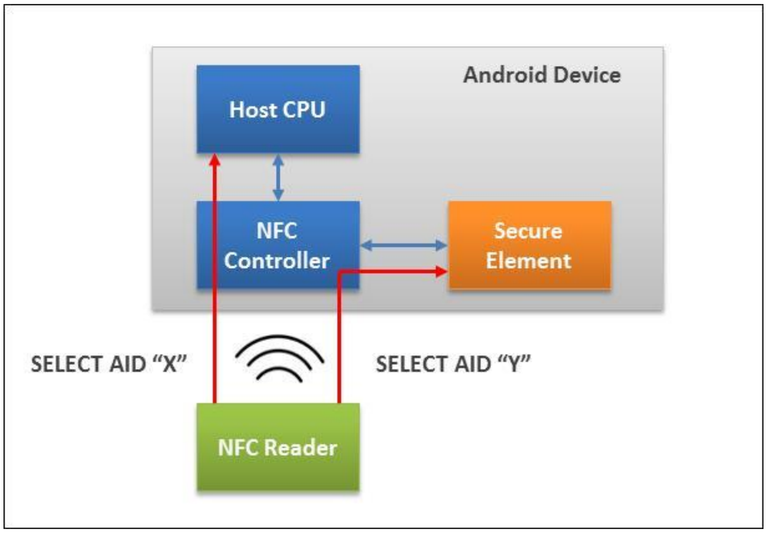

Figure 7. Routing Options for NFC Payment Transactions

AID-specific routing supports the coexistence of applications relying on HCE or the SE in the same device. This capability provides an opportunity for applications to share AIDs. The ability to do partial AID matching[24] is facilitated through the different NFC controllers.

Android KitKat 4.4 also supports the definition of a default route when an AID is registered in the routing table. When partial AID matching is not supported and an AID set in the routing table cannot be found, the default route is used automatically.

An application can be implemented to give users the ability to select an appropriate mode of payment when the HCE implementation supports multiple routing options. This option also allows multiple solution providers and issuers to participate and offer their payment solutions on the same mobile device.

3.4 Impact of HCE on the Payment Ecosystem

HCE and its endorsement by the payment networks represent an opportunity for issuers and solution-providers alike to create new payment solutions.

3.4.1 Independence from Secure Element Issuers

In the SE-based NFC wallet, the SEs are owned by OEMs or MNOs, and the SE is a valuable asset. This ownership model requires all participants to execute contractual agreements with the SE's owner for using the SE. The owner designates a space (i.e., security domain) inside the SE for each issuing bank and grants the issuing bank access to it. This operation is performed by connections through the TSM infrastructure of the issuing bank and SE issuer.

This model implies a triple dependency for issuing banks on the SE issuers:

1) Business dependency: Commercial agreements must be established for "SE rental."

2) Roadmap dependency: Deployment of the bank service to its customers is contingent on SE issuers massively deploying SEs to their customers (NFC SIMs for MNOs, or embedded SEs for OEMs), as well as implementing and integrating with the TSM infrastructure.

[24] Partial AID matching may be used depending on NFC controller memory requirements.

3) Brand dependency: In many cases, SE issuers (i.e., MNOs) define the user experience through their own branded wallet application, and bank issuers are exposed only as a sub-brand within the application, often side-by-side with their competitors.

HCE removes the role of the SE issuer and the associated dependencies for issuing banks.

3.4.2 Development and Deployment for Basic Services

With HCE, service providers can design, develop and deploy services in an autonomous way. Retailers can add NFC payment to their existing application (e.g., as in the case of Tim Hortons). Google's open access HCE implementation for Android apps frees developers and other members of the ecosystem from dependency on the owner of the SE. App developers anywhere can build apps to support custom contactless applications for loyalty, access, coupons, transit, and closed loop payments.

A host card emulation app is identified by a unique AID. HCE on Android supports the organization of AIDs into groups and categories. Transactions can be routed to the SE when appropriate; an app can be identified as a payment or non-payment app. Android HCE is built as a service model; it can be activated when a transaction takes place and execute in the background, rather than having to be started every time a transaction occurs. This feature can support multiple simultaneous use cases without requiring user intervention at every transaction.

3.4.3 POS Acceptance Process

Apps that use HCE to support contactless payment must comply with the same specifications as apps that rely on the SE. HCE therefore has no impact on the POS application. Payment brands can continue to use their current contactless specifications; where the credentials are stored is transparent to the POS. Communication with the POS is still based on the ISO/IEC 14443 standard and EMVCo specifications, Book D.[25]

As already mentioned, an HCE payment application is likely to use tokenized credentials rather than static payment credentials. In order to ensure that no modification is required in the acquiring infrastructure, the tokens, parameters, cryptograms and commands exchanged between the POS and the HCE application will have to keep the same format already in place for contactless bank cards and SE-based mobile applications.

3.4.3.1 CONSUMER PERSPECTIVE

For the consumer, the experience at the POS will depend on the specific HCE-based payment implementation approach.

For example, if the payment approach uses tokens, it is likely that one or more tokens will be stored in memory on the mobile device. When the consumer taps the phone at the POS, the phone must be powered so that the token can be retrieved and sent to the POS. If the token is valid, the transaction would process as a normal contactless, tokenized transaction. However, if the token has expired and no other token was available on the device, a network connection would be needed to download a new tokenized payment credential from the cloud. The tap may take longer or, if no network connection is available, the consumer may not have a valid payment credential to use.

With current tokenization specifications, tokens may not be able to be used with offline POS terminals; tokens must be authenticated in real-time. In an HCE implementation that uses tokens, the transaction would be rejected if the POS terminal is offline.

It is also important to note that additional user authentication could be required for HCE-based solutions.

[25] EMV Contactless Specifications, EMVCo, http://www.emvco.com/specifications.aspx?id=21

3.4.3.2 OTHER POTENTIAL USES

The use of HCE can enable a wide range of value-added retail applications that leverage the NFC interface, such as combining offers, coupons, or loyalty with payment.

Because communication between the app and the POS complies with the ISO/IEC 14443 standard, any value-added applications should use the same protocol to optimize transaction times. Note, however, that there are no standards for value-added services. Implementations are provider specific, although certain companies and organizations such as GSMA are working on specifications; some specifications are currently available under license.

3.4.4 Hardware Security vs. Risk Mitigation

The security model of an SE implementation relies on its strong hardware security and the tamper-proof SE chip. The level of protection and assurance that the SE provides allows the payment brands and bank issuers to provision a payment credential to the mobile device, exactly like the ones that are loaded on physical cards during the production process. Once stored in the SE, the credential may be used for payment until its expiration date, or until the user decides to deprovision it.

With an HCE implementation, the equivalent approach would consist of storing the same payment credential inside the HCE service. However an application running on the mobile OS is much more exposed to malicious attacks than an applet in the SE, and the risk associated is too high for payments. To enhance security, HCE implementations may use white box cryptography and software tamper proofing and be supplemented by techniques such as device fingerprinting.

However, a practical HCE payment implementation may also need to incorporate risk mitigation techniques, such as tokenization, device fingerprinting, geolocation, and risk-scoring prior to authorization. Strong user authentication or storage of credentials in a TEE or SE could help to further reduce the risk.

Additional information on security approaches is included in Section 4.

3.4.5 Summary of HCE Impact on Payments Ecosystem

The international payment brands have defined specifications supporting HCE-enabled NFC payment applications. The specifications support both the contactless magnetic stripe data protocol and contactless EMV. NFC mobile payment transactions carried out using HCE specifications are therefore likely to be considered card-present transactions, depending on payment network policy. However, as described in Section 3.4.3.2, the customer experience may not be comparable to what is experienced when credentials are stored in the SE and other risk mitigation approaches may be required to bolster the security of HCE payment implementations.

4 Security Considerations

When an app uses HCE, communications with the contactless terminal are no longer routed to the SE but through the NFC controller to the mobile device’s host CPU on which the app is running. This change introduces certain risks.

Communication between the NFC controller and the HCE-enabled app can be spied on by malware applications. Malware applications can attack the operating system, a risk which is exacerbated when the handset is compromised by exploiting, rooting or jailbreaking. The malware itself may also be able to exploit, root or jailbreak the device, or spoof the user into initiating such actions. In addition, denial of service attacks can take place if routing is changed by a malware application. More generally, cloud storage and backup servers can be attacked, as can credentials stored in applications that are used to gain access to cloud storage and backup servers.

Various measures can enhance the security of HCE:

- White box cryptography
- Tamper proofed software
- Biometric factors
- Device identity solutions
- Security frameworks/trusted execution environment
- Encryption
- Tokenization
- Additional security provided by an SE

4.1 White Box Cryptography

White box cryptography is a form of obfuscation for deterministic algorithms, and it is usually applied to cryptographic algorithms. It is used to prevent the exposure of secrets (usually keys) in memory or in code. Generally a white box implementation of cryptography turns a cipher into a robust form where the secret is combined with the code such that it cannot be easily derived or distinguished, but can be used in place to create an obfuscated boundary for processing.

4.2 Tamper-Proofed Software

Tamper proofing of software (also known as tamper detection, anti-tamper or tamper resistance) is the addition of software security to software in order to make it harder for an attacker to change or modify the software statically or dynamically. Typically expressed in the form of runtime integrity checking, most systems also include other defenses to make tampering or reverse engineering harder such as obfuscation, breakpoint defenses, anti-debug and other measures. Upon detecting an attack, tamper-proofed systems generally produce a response which usually makes a program malfunction, fail to operate, or record and communicate the attack.

4.3 Biometric Factors

Biometric factors can be used to strengthen user authentication for HCE-enabled applications in addition to other means of authentication. One advantage of using biometric factors is its relative user friendliness, compared (for example) to requiring a multitude of passwords. Gartner predicts that by 2016, 30 percent of organizations will be using biometric data for authentication.

Currently, three types of biometric factors can be used:

- Fingerprints
- Facial recognition
- Voice recognition

Fingerprint readers have been included in laptops and other devices for some time. Samsung introduced a fingerprint reader with its new Galaxy S5, which also supports NFC.

Facial and voice recognition are implemented in various mobile phone models and can be used at the application level.

If biometrics are used, the privacy and security of the biometric data must be considered in application implementation.

4.4 Device Identity Solutions

Device identity solutions authenticate handsets to online services. A number of solutions are available to support device identity and can provide an additional layer of security for HCE-based applications.

An example of an approach for device identity is the Fast Identity Online (FIDO) Alliance specification.

FIDO protocols use public key cryptography techniques to provide online authentication. When using an online service, a user's device creates a new key pair, retaining the private key and registering the public key with the online service. The user's device authenticates through signing off with the private key, which can only be unlocked locally on the device through secure mechanisms such as swiping a finger (biometrics) or entering a PIN.

The FIDO approach supports a range of different technologies that can co-exist, including (for example) tokenization and one-time password (OTP) solutions.

Samsung has built-in biometric security. PayPal is the first service provider to use the fingerprint verification functionality in the recently launched Galaxy S5 handset with FIDO Ready software.[26] (FIDO Ready products are based on the draft FIDO technical specification.[27])

4.5 Security Frameworks/Trusted Execution Environment

The trusted execution environment (TEE) is a secure area in the main processor or coprocessor of a mobile device in which data can be stored and processed.[28] The TEE can support safe execution of authorized security software (trusted applications) in a trusted environment.

The TEE is composed of software and hardware, offering protection against software attacks originating from the rich operating system (Rich OS) in a mobile device. The TEE assists in the control of access rights and houses sensitive applications that need to be isolated from the Rich OS.

The TEE can work with the SE to provide protection. For example, the TEE can provide a secure interface to transmit a PIN stored in the SE or filter access to applications stored in the SE.

Since the TEE runs its own operating system, it is not affected if the handset's main operating system is compromised. For HCE-enabled applications, the TEE can provide an additional level of security:

- **PIN/password entry.** The TEE will allow additional protection of the HCE solution by allowing secure input of a PIN or password. (The TEE has the ability to obtain a completely separate secure input from the input of the mobile device that cannot be intercepted by malware on the mobile device OS. This allows local stored tokens to be unlocked, the individual to be authenticated to the cloud part of the HCE

[26] "The FIDO Alliance Announces First FIDO Authentication Deployment – PayPal and Samsung Enable Consumer Payments with Fingerprint Authentication on New Samsung Galaxy S5," FIDO Alliance press release, February 24, 2014, https://fidoalliance.org/news/item/the-fido-alliance-announces-first-authentication-deployment-paypal-samsung.

[27] https://fidoalliance.org/specifications/download/.

[28] Trusted Execution Environment Guide, Global Platform™, http://www.globalplatform.org/mediaguidetee.asp.

solution, or even the input of a PIN that is then transmitted to the terminal that the HCE-enabled mobile device is interacting with.)

- **Secure storage of credentials.** The TEE allows secure storage of keys and implements the main cryptographic operations directly within the boundaries of the secure execution environment. This allows, for example, the storage of tokens for payment applications and offers enhanced protection against exploits compared to the standard mobile device OS.
- **Secure transfer protocol endpoint.** Since the TEE allows the loading of trusted applications (TAs) and related cryptographic material, it is possible to let a cryptographic secure channel from the terminal end in the TEE. This means that the commands (APDUs) transferred from the terminal via the contactless interface and HCE are encrypted and transmitted all the way into the TEE, and hence can be protected for integrity and privacy. In addition, a second secure transfer channel can be employed between the TEE and a cloud application. In this manner keys and data are only ever visible in clear in the TA, allowing a higher level of protection compared to the applications running in the mobile device OS. It is important to note, however, that the TEE may not have the SE's tamper resistance, depending on the implementation.

4.6 Encryption

Encryption ensures sure that data is not transmitted in plain text. Siphoning cleartext data has been the culprit in data breaches in card-present and card-not-present payment environments, when cards are swiped or inserted, or when data is entered into Web-based forms (using man-in-the-middle attacks). HCE data can be encrypted and data can be stored within the applications.

End-to-end encryption (E2EE) or point-to-point encryption (P2PE) ensures that data is encrypted at the reader and protected during transmission. The purpose of E2EE or P2PE is to mitigate the risk of data interception when data are transmitted.

Different encryption methods using different key types may be applicable. The Payment Card Industry Security Standards Council (PCI SSC) has developed standards for P2PE that dictate secure management of cryptographic keys for payment processing.[29] E2EE or P2PE can be applied to the data as well as to the connection through which the data are being transmitted.

Encryption can be applied in combination with tokenization for payment applications (discussed in the next section and in Section 3.2.2). Cardholder data (most critically the PAN) can be encrypted and the encrypted PAN then used for tokenization, with tokens replacing the PAN.

4.7 Tokenization

Tokenization is the process of substituting a random value for a high value credential (e.g., a PAN or Social Security number), thereby creating a low value equivalent. Tokenization can be used to mask the identity of a card.

While tokenization itself is not a new concept, recent data breaches have increased both awareness of and the need for tokenization as a mechanism for protecting payment credentials against fraud and counterfeiting. Various commercial tokenization implementations are already available and industry bodies have introduced new tokenization standards.

EMVCo recently announced a tokenization specification for payments.[30] EMVCo's tokenization can be domain specific and includes cryptograms that can isolate HCE-based NFC use cases. This approach can prevent use of the token in other payment channels.

[29] "P2PE Hardware/Hardware Solution Requirements and Testing Procedures," PCI Security Standards Council, June 2013, https://www.pcisecuritystandards.org/security_standards/documents.php.

[30] "EMV Payment Tokenisation Specification – Technical Framework," EMVCo, March 2014, http://www.emvco.com/specifications.aspx?id=263.

In addition to EMVCo, X9 and PCI both have efforts to develop standards for tokenization.[31]

4.8 Secure Element

HCE does not dictate where to store data. HCE enables an NFC controller to communicate directly with applications running on the host CPU. Data can be stored in the cloud or in the SE. Even though HCE enables NFC contactless technology without an SE, a hybrid model that uses the SE in combination with a cloud-based solution is possible.

The SE can store payment and other data securely. SE security is guaranteed through the use of cryptographic keys (symmetric and asymmetric). The security of the SE can be enhanced through the TEE, which can interpose between the Rich OS and the SE and allow only trusted applications to access the SE. (The TEE is described in Section 4.5.)

It should be noted that this is still riskier than the situation where both sensitive data and programs (applets) are deployed on secure element hardware separate from host OS. Furthermore, using an embedded SE in combination with the NFC controller, where an additional parameter indication of 'contacted'/wired mode versus 'contactless'/virtual mode interface, can be very useful to monitor and prevent remote relay attacks. Most NFC controllers have secure firmware, which would make it difficult to replace the firmware with rogue firmware where the NFC interface to the SE or HCE can be compromised.

[31] Additional information can be found in the Smart Card Alliance white paper on EMV, tokenization and encryption that can be found at http://www.smartcardalliance.org.

5 Non-Payment Use Cases

HCE can open up application development to a variety of non-payments use cases. It is important to remember that the infrastructure with which the HCE credential is used must be able to support HCE-based transactions. (For an example, see Section 3.4.3 for how the user experience could be impacted for a payment application.)

This section provides examples of HCE use cases for other applications, including: promotions, advertising, and coupons; loyalty programs; transit ticketing and payment; and physical access control.

5.1 Promotions, Advertising and Couponing

Applications that control interactions with advertising are a logical use of HCE. While these interactions can have a non-cash value, they may be unlikely to be valuable enough to warrant attack. Such interactions are also less likely to require speed, as they may not take place at the POS.

One example of such a use case would be a campaign that provides consumers with coupons to redeem at the POS. HCE could be used to assure that the consumer qualifies for the benefit by verifying each required interaction.

It is important to note, however, that there are still serious brand reputation risks since mobile devices and applications are vulnerable to hacking that could subvert one of these applications (e.g., if there is malware on the phone or at the POS).

5.2 Loyalty

Loyalty is an area in which interactions with a customer have a non-cash value, so providing a degree of security is beneficial. An enterprise could use HCE to support a rewards program in conjunction with other transactions. For example, a payment could be followed by a loyalty reward. A POS system could trigger the loyalty reward.

5.3 Transit

Most transit scenarios require an accredited secure standard or specification, such as MIFARE, to protect the high value of the tickets and ensure that the related identity management tokens are stored and accessed securely. In addition, MIFARE Classic does not support the use of simple AIDs, which is a core component of an HCE implementation. MIFARE DESFire has usage modes that do support AIDs, but much of the infrastructure in place does not use an AID for initiating a communication with MIFARE DESFire, instead using the native commands; this presents a limitation for using HCE for transit applications in the current incarnation.

HCE would also face difficulties in transit implementations that require throughput, which is critical to the simple and safe use of NFC at crowded turnstiles. Users must first wake up the mobile device. It is also difficult to establish a consistent transaction timeframe for HCE transactions since the speed depends strongly on the workload of the host CPU and applications running in parallel on the phone.

In addition, an NFC application that relies on the SE can provide battery-low and battery-off modes, which HCE cannot. Finally, connectivity in many transit locations can be challenging when real-time debiting of accounts is a necessity.

5.4 Physical Access Control

A wide variety of security options are currently deployed to support physical access control. Some access privileges must be protected by very strong security; some privileges require certification by government agencies. Other access privileges, such as access to a common area in a building, require lower levels of security (e.g., a closed loop access control solution).

As with transit applications discussed in section 5.3, physical access control systems are based on contactless technology that may or may not be compatible with NFC.

- Current NFC solutions that support MIFARE represent significant compatibility challenges with HCE, as discussed in section 5.3.
- Newer access control solutions that are based on NFC standards can leverage HCE on supported devices.
- Many legacy physical access control systems are not based on ISO/IEC 14443 and are not compatible with NFC.

HCE can add value to an NFC-based access control solution because it has the potential to allow more widespread adoption through a ubiquitous user experience across the entire market. It opens up new market segments where 'throw away' access control tokens are being used such as the hospitality market (hotels) and potentially the residential market. For some use cases that require higher security, an HCE-based access control solution can be complemented with additional security layers such as using the device PIN/fingerprint scanner or end-to-end encrypted tokenization to protect against attacks. Some applications may also require a secure channel for communication between the reader/terminal and the mobile device.

6 Conclusions

Of the three modes NFC offers for mobile devices, card emulation has been the most popular, and also the most controversial, mode, due to the need to access the secure element that is owned and controlled by another party. HCE significantly changes card emulation implementation requirements and introduces entirely new business plan considerations for service providers and issuers wishing to use their credentials for NFC use cases.

Along with the greater flexibility HCE offers for service providers and issuers, comes advantages and trade-offs to the traditional SE model and accompanying (required) ecosystem. Some advantages include more direct control and fewer dependencies on other ecosystem players. Some disadvantages include a less secure implementation and, possibly, a degraded end user experience in some cases. The list of advantages and trade-offs will change as more HCE-based solutions are deployed, tested and used in commercial practice.

In summary, NFC continues to gain strong industry support from an increasing number of suppliers, manufacturers and handset models; however, HCE currently is only commercially supported on Android and Blackberry, and specifications still need to mature and be harmonized across OS vendors. While HCE is not the 'silver bullet' many would like to have, it has far-reaching implications for the industry in general. Further, HCE introduces an attractive option and welcome solution for those service providers and issuers whose business models do not require, or cannot thrive within, a traditional secure element-based implementation.

7 Publication Acknowledgements

This white paper was developed by the Smart Card Alliance Payments Council to provide an educational resource on host card emulation (HCE), describing what it is, how it's used, how it compares with the secure element (SE)-based card emulation approach, what key considerations are for payment applications, and what security aspects should be considered for HCE-enabled applications.

Publication of this document by the Smart Card Alliance does not imply the endorsement of any of the member organizations of the Alliance.

The Smart Card Alliance wishes to thank the Payments Council members for their contributions. Participants involved in the development of this white paper included: ABnote Group; Advanced Card Systems Ltd.; BetterBuyDesign; Capgemini USA Inc.; CH2M Hill; Clear2Pay; CorFire; Cubic Transportation Systems; Discover Financial Services; Eid Passport Inc.; First Data Corporation; Fiserv Inc.; Gemalto; Giesecke & Devrient; HID Global; HP Enterprise Services; Identiv; Ingenico; Initiative for Open Authentication (OATH); INSIDE Secure; Intercede; IQ Devices; Isis; MasterCard; Morpho; NXP Semiconductors; Oberthur Technologies; OTI America; Thales e-Security; Underwriters Laboratories (UL); Valid USA; VeriFone.

The Smart Card Alliance thanks **Sree Swaminathan**, First Data, who led the project and the Council members who participated in the project team to write the document, including:

- **Brent Bowen**, Valid USA
- **Tatiana Castro**, Oberthur Technologies
- **Greg Coogan**, Morpho
- **Stu Cox**, Giesecke & Devrient
- **Ian Hermon**,Thales e-Security
- **Gonda Lamberink**, UL
- **Peter Lee**, CorFire
- **Don Malloy**, OATH
- **Cathy Medich**, Smart Card Alliance
- **Bob Merkert**, Advanced Card Systems
- **Akif Qazi**, Discover
- **Lakshmi Ramanathan**, Discover
- **JC Raynon**, VeriFone
- **John Rego**, OTI America
- **Steve Rogers**, IQ Devices
- **Sherif Samy**, UL
- **Rajesh Sharma**, INSIDE Secure
- **Brian Stein**, CH2M Hill
- **Sree Swaminathan**, First Data
- **Sastry Yeleswarapu**, Capgemini

The Smart Card Alliance thanks the Council members who contributed to the review of the document, including:

- **Philip Andreae**, Oberthur Technologies
- **Deborah Baxley**, Capgemini
- **Don Campbell**, Eid Passport
- **Peter Cattaneo**, Intercede
- **David deKozan**, Cubic
- **Jim Ellis**, ABnote
- **Michael Esser**, Fiserv Inc.
- **Jeff Fonseca**, NXP Semiconductors
- **Allen Friedman**, Ingenico
- **Philip Hoyer,** HID Global
- **Jack Jania**, Gemalto
- **Umesh Kulkarni**, Clear2Pay
- **Robert Leonard**, MasterCard
- **Don Malloy**, OATH
- **Oliver Manahan**, MasterCard
- **Pedro Martinez**, Gemalto
- **Russ Kent**, HP Enterprise Services
- **Andrew McLennan**, Metaforic/INSIDE Secure
- **Sadiq Mohammed**, MasterCard
- **Steve Mott**, BetterBuyDesign
- **Tony Sabetti**, Isis
- **Lars Suneborn**, Identiv
- **Bart van Hoek**, UL
- **Tom Zalewski**, CorFire

Trademark Notice

All registered trademarks, trademarks, or service marks are the property of their respective owners.

Smart Card Alliance
Mobile & NFC Council

About the Smart Card Alliance Mobile & NFC Council

The Smart Card Alliance Mobile and NFC Council was formed to raise awareness and accelerate the adoption of payments, loyalty, marketing, promotion/coupons/offers, peer-to-peer, identity, and access control applications using NFC. The Council focuses on activities that will help to accelerate the practical application of the technology, providing a bridge between technology development/specification and the applications that can deliver business benefits to industry stakeholders.

The Council takes a broad industry view and brings together industry stakeholders in the different vertical markets that can benefit from mobile and NFC applications. The Council collaborates on: educating the market on the technology and the value of mobile and NFC applications; developing best practices for implementation; and working on identifying and overcoming issues inhibiting the industry.

8 References

"BBVA introduces HCE-based mobile payments," Finextra, June 30, 2014, http://www.finextra.com/news/fullstory.aspx?NewsItemID=26217

EMV Contactless Specifications, EMVCo, http://www.emvco.com/specifications.aspx?id=21

"EMV Payment Tokenisation Specification – Technical Framework," EMVCo, March 2014, http://www.emvco.com/specifications.aspx?id=263

"The FIDO Alliance Announces First FIDO Authentication Deployment − PayPal and Samsung Enable Consumer Payments with Fingerprint Authentication on New Samsung Galaxy S5," FIDO Alliance press release, February 24, 2014, https://fidoalliance.org/news/item/the-fido-alliance-announces-first-authentication-deployment-paypal-samsung

FIDO Alliance specifications, https://fidoalliance.org/specifications/download/

"Google Wallet ends support for physical secure elements," *NFC World*+, Mar. 17, 2014, http://www.nfcworld.com/2014/03/17/328326/google-wallet-ends-support-physical-secure-elements/

Google Wallet web site, http://www.google.com/wallet

"Isis Reports 600K New mWallet Downloads In The Last Month," May 14, 2014, PYMNTS.com, http://www.pymnts.com/news/2014/isis-reports-a-600k-new-mwallet-downloads-in-the-last-month/?utm_source=May+15%2C+2014&utm_campaign=US+NL+May+15%2C+2014&utm_medium=email

Isis web site, http://www.paywithisis.com

"MasterCard to Use Host Card Emulation (HCE) for NFC-Based Mobile Payments," MasterCard press release, Feb. 19, 2014, http://newsroom.mastercard.com/press-releases/mastercard-to-use-host-card-emulation-hce-for-nfc-based-mobile-payments/

"A Message from Michael Abbott: Embarking on a New Brand," July, 7, 2014, http://news.paywithisis.com/2014/07/07/isis-wallet-rebranding/

"NFC Forum Statement Regarding Host Card Emulation (HCE)," March 20, 2014, http://nfc-forum.org/newsroom/nfc-forum-statement-regarding-host-card-emulation-hce/#_ftn2

NFC Forum web site, http://www.nfc-forum.org

"NFC phones: The definitive list," *NFCWorld+*, May 11, 2014, http://www.nfcworld.com/nfc-phones-list/

"P2PE Hardware/Hardware Solution Requirements and Testing Procedures," PCI Security Standards Council, June 2013, https://www.pcisecuritystandards.org/security_standards/documents.php

Smart Card Alliance NFC Resources, http://www.smartcardalliance.org/pages/smart-cards-applications-nfc

Smart Card Alliance web site, http://www.smartcardalliance.org

"Smartphones accounted for 80% of the NFC devices shipped in 2013," ABI, *NFCWorld+*, Jan. 8, 2014, http://www.nfcworld.com/2014/01/08/327447/abi-smartphones-accounted-80-nfc-devices-shipped-2013/

"Tim Hortons launches NFC payments service using Host Card Emulation," NFC World, December 13, 2013, http://www.nfcworld.com/2013/12/13/327339/tim-hortons-launches-nfc-payments-service-using-host-card-emulation/

"Trusted Execution Environment Guide," Global Platform, http://www.globalplatform.org/mediaguidetee.asp

"Visa to Enable Secure, Cloud-Based Mobile Payments," Visa press release, February 19, 2014, http://investor.visa.com/news/news-details/2014/Visa-to-Enable-Secure-Cloud-Based-Mobile-Payments/default.aspx

9 Appendix A: Secure Element and Host Card Emulation Implementation Considerations

The following table summarizes areas of consideration for secure element and host card emulation implementations, focusing on areas where there are differences between the approaches.

Areas of Consideration	Secure Element	Host Card Emulation
User Experience		
Battery consumption	Battery consumption can be less, based on implementation.	Since host OS interaction is required, higher power consumption can be expected.
Battery-off function	Possible to work in low or no battery modes.[32]	Requires battery power.
Mobile devices supported	Secure elements and apps work on NFC-enabled mobile devices.	Currently supported on Android 4.4 or higher and Blackberry OS 7 or higher. Specific app versions are required for each OS.
Mobile device change	Managed through the TSM.	Managed by the solution provider.
Latency for accessing credential during a transaction	Credentials are readily available as they are stored inside the SE locally.	A pure cloud-based storage solution is problematic due to latency. Credentials (either static or tokens) must be already in the device when the NFC transaction takes place.
Life cycle management	Only needed for static credentials and apps. Defined by GlobalPlatform standards. Same for all payment brands for payment applications.	Needed for apps, static credentials and tokens (if required by use case to mitigate risk). Life cycle management solution is proprietary for each payment brand for payment applications.
Maturity of deployments	Many commercial deployments around the world over the last 3 – 4 years.	A few commercial solutions.
Device connectivity to mobile network or WiFi	Required only for provisioning of credentials. Transactions do not require device to be connected to the network.	Required for provisioning of credentials. Transactions may require network connectivity to mitigate security risks (e.g., if solution requires tokenization, connectivity would be required to acquire/refresh tokens).
Online/offline acceptance infrastructure for transactions	Both online/offline modes are supported.	Significant security challenges exist to support offline mode.
Implementation		
AID conflicts	Strict AID rules.	Can share and reuse AIDs between client mobile applications.

[32] Battery-power-low mode. The mobile device is not bootable. The device does not have enough power to run the operating system. The user is unable to switch on the mobile device. Screen, keyboard and other functions are not available. The mobile device, however, should supply full power to the NFC chip and secure element as long as some power from the battery is available.

Areas of Consideration	Secure Element	Host Card Emulation
Approach for provisioning and managing credentials	Managed through TSM infrastructure for SE-based applications; does not require tokenization for sensitive credentials.	Managed through the application. Complexity grows when risk mitigation techniques need to be introduced.
Co-existence with other credential storage form factors	SE can co-exist with HCE, depending on OS vendor implementation.	HCE can co-exist with SE, depending on OS vendor implementation.
Cost	Determined by the owner of the SE. Main cost components are typically SE rental fees and TSM infrastructure.	Service provider does not need to negotiate SE rental fees. IHCE implementation still needs an infrastructure to provision credentials into the mobile device. If the use case requires tokenization to mitigate risk of exposure of the credentials, cost of the tokenization infrastructure must be factored in.
Credential storage options	Stored inside SE and highly tamper proof.	Can be stored in the cloud or in the host CPU with the use of OS-based cryptography; can be stored in an SE or the TEE on the mobile device.
Memory limitation	SE chip may have memory limitations and is a consideration.	No memory limitation.
Provisioning infrastructure	TSM is necessary to create security domains, provision apps and credentials, and provide lifecycle management.	Provisioning infrastructure is required to manage credentials and tokens. Requires active lifecycle management for credentials/tokens.
Real-time changes to applications	Possible to change applets through SE infrastructure. GlobalPlatform specifications are being used for management.	Same process as used for updating any mobile app.
Secure element issuer (SEI)	SEI participates in both business model and technical implementation, as it owns the SE that will hold the application/credentials and must allocate a partition to each service provider wanting to use the SE.	No SEI is involved but infrastructure needs to be implemented to mitigate security risks for credentials stored with the application in the mobile device.
Standards and Specifications		
Certification criteria	Very well defined by all payment brands for payment applications.	Currently not defined by any payment brand for payment applications.
Maturity of the technology	Backed by strong and mature standards.	Mobile OS implementation of HCE is immature.
Maturity of the specifications	Backed by strong and mature standards for the SE and applications.	Specifications supporting HCE are evolving and need to be harmonized across OS vendors. Proprietary payment specifications are either immature or unreleased.
Standards and interoperability	Standards support interoperability at provisioning, secure element, NFC radio, mobile OS, and payment and transit application levels.	Only EMVCo tokenization and two proprietary payment brand specifications are available for interoperable payment applications.

Areas of Consideration	Secure Element	Host Card Emulation
Standards and specifications	GlobalPlatform, ETSI, ISO/IEC 14443, ISO/IEC 7816, ISO/IEC 18092, MIFARE, and all global payment brands.	ISO/IEC 14443, ISO/IEC 7816, ISO/IEC 18092 and two global payment brand specifications.
	Security	
Cryptography	GlobalPlatform session encryption keys are employed to protect the payload being provisioned to and stored on the SE.	Application-specific.
Risk of malware attack	None. SE is isolated from the mobile OS during transaction.	The HCE app resides in an open and connected mobile OS, and is subject to malware attacks like any other mobile app on the phone.
Security	Highly secure due to protection of data inside tamper-proof module.	Lower security level due to software-based storage. Risk mitigation can be achieved by tokens, additional authentication methods and/or software-based security approaches.
Tokenization of credentials	Possible, but not required even for open loop payments. Credentials are safely stored in the SE.	Likely to be required for open loop payment. It may be advisable for other use cases dealing with sensitive credentials. Could present customer service issues if tokens change per transaction.
	Use Cases	
Closed loop payment	Requires similar infrastructure to open loop.	HCE services for closed loop are straightforward to implement.
Card-present/card-not-present status for face-to-face payment transactions	Payment transactions are considered card-present.	Will be based on payment network policy.
EMV compatibility for payment	EMV contactless compatible, with certification programs enforced.	Can be EMV contactless compatible; currently no certification available.
MIFARE support	Yes.	Currently not available.
Multiple wallet integration	Multiple wallet integration in single device can be complex; OS determines how multiple wallets are handled.	Multiple wallet integration in single device can be complex; OS determines how multiple wallets are handled.
Open loop payments	Supported with credential in SE.	Likely to be supported with tokenized credential.
Transit, loyalty, access applications	Solution providers must coordinate with SE issuers. Can be compatible with most deployed acceptance infrastructures.	Solution providers can implement solutions independently. Backward compatibility issues with deployed acceptance infrastructure must be addressed.