Mobile payments using Host Card Emulation with NFC:
security aspects and limitations

Authors
Shana Micallef, MSc (Royal Holloway, 2017)
Konstantinos Markantonakis, ISG, Royal Holloway

Abstract

Smartphones, with NFC capabilities, are gradually becoming one of the preferred method over credit cards in contactless payments. However, unlike payment cards, smartphones are an easier target for hackers. This raises concerns of security threats and data privacy issues, particularly when the token is stored on a smartphone, and when such data is communicated through less secure networks such as the Internet. In this article, we present a set of risks associated with using smartphones for contactless payment transactions. [a]

[a] This article is published online by Computer Weekly as part of the 2018 Royal Holloway information security thesis series `https://www.computerweekly.com/ehandbook/Mobile-payments-using-Host-Card-Emulation-with-NFC`. It is based on an MSc dissertation written as part of the MSc in Information Security at the ISG, Royal Holloway, University of London. The full thesis is published on the ISG's website at `https://www.royalholloway.ac.uk/research-and-teaching/departments-and-schools/information-security/research/explore-our-research/isg-technical-reports/`.

Introduction

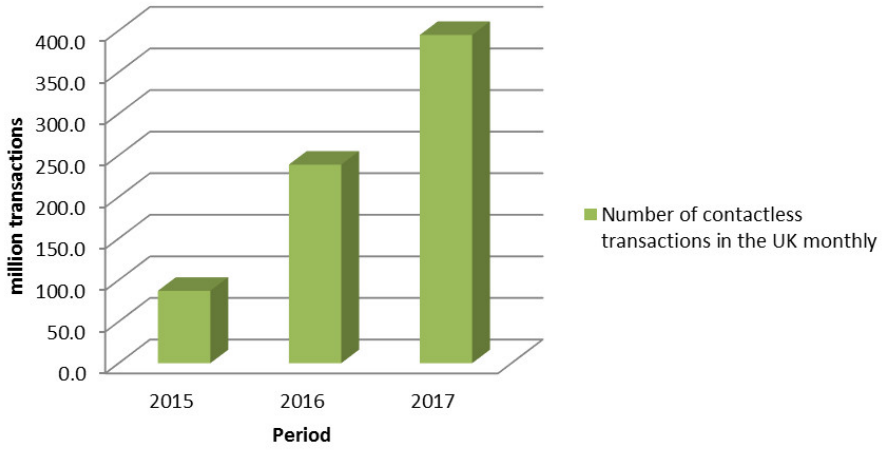
One of the recent forms of payment technologies introduced by EMV, the global standard for credit and debit cards, relies on Near Field Communication (NFC) to transfer transaction data between a Point of Sale (POS) and the credit card. This form of payment is known as contactless payment. Since smartphones have NFC chips embedded in their hardware, smartphones can emulate a payment card in contactless payment. Such process is known as Host Card Emulation (HCE). During a transaction, data from the NFC chip is routed to a special software application, known as a "Wallet". The wallet application is responsible for storing, to an extent, personal data traditionally stored in a secure chip in the credit card.

Earlier attempts to use smartphones as credit cards utilized the SIM card as a secure element (SE). However, this had disadvantages, mostly because the mobile network operator has created some barrier to access the SE. Thus, in October 2013 Google introduced HCE in the Android Operating System allowing a Wallet App to communicate with the NFC hardware and thereby emulating a contactless card. To mitigate the risks of storing personal data on the smartphone the industry came up with different techniques and technologies such as storing information on the cloud and replacing credit card numbers with limited-use tokens, a process known as *tokenization*.

What is HCE?

HCE is essentially an extension to the EMV contactless technology. EMV contactless cards use Near Field Communication to communicate with the payment terminal. HCE implementations vary between different mobile platforms (e.g. Android versus Windows phone) but in essence they all follow the same procedure.

Figure 1: Number of contactless transactions in the UK (average monthly). *Source of data:* `http://www.theukcardsassociation.org.uk/contactless_contactless_statistics/index.asp`



Card Enrolment

Prior to any transactions, the user has to "enrol" a payment card in the smartphone. There are different ways for this to occur. The smartphone can read the data from the payment card through NFC or the data could be entered manually by the user or by taking a photo of the card. During enrolment, the smartphone also registers the card's Application Identifier (AID), a number that identifies the card's vendor (Visa, MasterCard, etc.). During this process a smartphone may also go online to verify the cardholder and to download data related to that particular card.

Contactless Payment

When a smartphone is tapped on a contactless Payment Terminal (POS), the NFC hardware informs the kernel of the pending transaction and provides information about the transaction including the AID that tells the kernel which card to use for that particular transaction. The kernel will then start or resume the wallet application for that card type. From then onwards, the wallet application is given access to the NFC hardware and the application will communicate directly with the POS to provide information required by the POS for the transaction to occur.

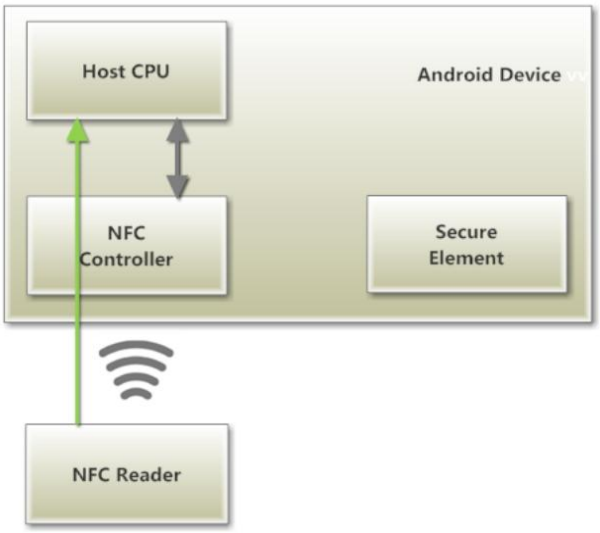
Figure 2 outlines the block diagram of HCE implementation in Android. In HCE, the wallet application runs as a service process and the transaction can occur in the background without any need for user interaction apart from the need for the smartphone to be unlocked. Some implementations allow the transaction to occur even when the smartphone is in standby (locked with screen off) mode.

Currently any entity can develop a wallet application. Some entities, such as Google have come up with wallets, such as Android Pay, to support different types of cards. Banks, like Barclays, have developed their own wallet application to support their own set of cards. However, payment networks (e.g. Visa) and wallet providers (e.g. Android Pay) are only a subset of the stakeholders involved in the HCE payment ecosystem as companies are constantly introducing new roles and services within this industry.

Tokenization

HCE brought about a more open mind-set in the payment industry when compared to the traditional plastic credit cards, in which the card issuer had, to an extent, total control over the hardware used.

Figure 2: Block diagram of HCE in Android. (*Source:* Smart Card Alliance, "Host Card Emulation (HCE)", [Online]. Available: `http://www.smartcardalliance.org/downloads/HCE_Webinar_FINAL_061815.pdf`.)



With HCE, consumers can use any phone with NFC, running any operating system (OS) and OS version that supports HCE, and using wallet applications from any entity that decides to develop one. Therefore, cyber-attacks to steal or manipulate sensitive card data became an issue and hence, the industry started to focus on measures that reduce the impact of such data theft through a process known as *tokenization*.

Using tokenization, account numbers, known as PANs (the 16 digit number) are translated into tokens which are essentially random numbers that expire and become useless after some set time according to some rules. During a transaction's clearing stage, the transaction is de-tokenised and the transaction can proceed to the card Issuer for authorization. The benefit of this technique is that tokens can have a limited value (e.g. used once only) or tied to a particular condition (e.g. to be used in a particular country/region) therefore, if stolen, its value is limited.

Stakeholders in HCE

Card Issuers and Acquirer

Card Issuers are payment institutions, typically banks that issue payment cards to their customers. Acquirers can be payment institutions or third parties that process payment transaction for merchants (e.g. shops) through the payment network.

Payment Network Providers

Payment Network Providers (e.g. VISA) manage and facilitate the flow of transactions, the clearing and settlement of card payment transactions between acquirers and Issuers.

Token Service Providers (TSP)

Entities such as issuers, the payment network or an independent third party can assume the role of a TSP. Their aim is to provide a tokenisation service by:

- Generating tokens for card holders to be stored on their smartphone.
- During a transaction:
  - Ensuring that the token is valid subject to its binding conditions (expiry, location, etc.).

– Translating the token back into the Payment Card Number (PAN) it represents.

Some implementations make use of another role known as the Token Requestor (TR). The TR's role is to store the token and the Unique Derived Key (UDK) from which Limited Use Keys (LUK) are generated. The TR would provide tokens and LUKs to the smartphone, keeping the UDK stored safely in its storage.

Wallet Application Providers

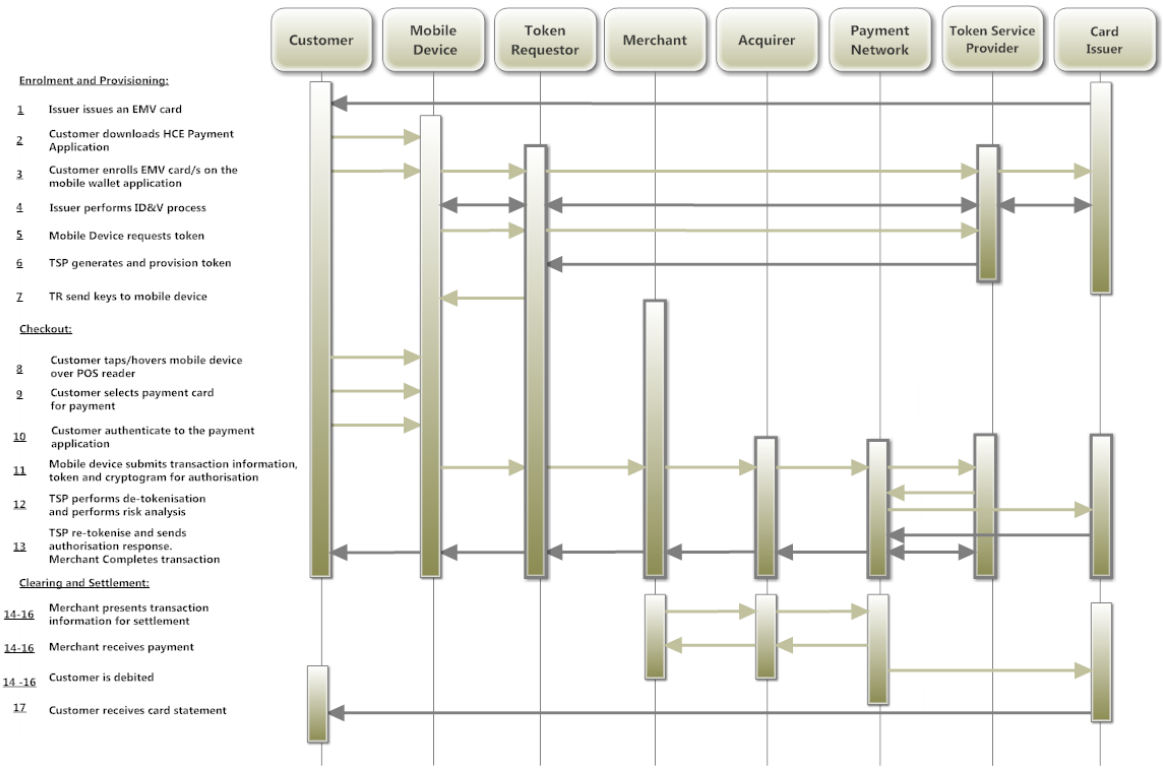
Wallet apps can be developed by issuers (e.g. Barclays), smartphone vendors (e.g. Samsung Pay), smartphone OS providers (e.g. Google - Android Pay) or third parties. The role of the wallet application provider is to develop and maintain the application that stores the details of the card and emulates the card during a transaction.

Contactless Payment Process

Figure 3 outlines the interaction between all the stakeholders to affect a contactless payment transaction. The process is split into three parts:

1. *Enrolment and provisioning*. Enrolment is done once to enrol a card in the wallet application while provisioning of tokens can occur periodically to download new tokens on the device.
2. *Checkout*. The cardholder uses the smartphone to pay for a product or service.
3. *Clearing and settlement*. The acquirer processes the payment for the merchant to receive the money.

Figure 3: Interaction between stakeholders during a contactless payment



The whole process is summarized below:

Enrolment and provisioning

1. The issuer issues a credit card to the cardholder. Typically the payment card is provisioned with application identifiers known as AIDs, customer specific data, specific transaction rules such as cardholder verification method, online/offline authorization and authentication and other security information according to the issuer's policy.
2. The cardholder downloads and installs a wallet application.
3. The cardholder then registers the card in the wallet application, either by using the camera to capture the card information or entering it manually. Multiple payment cards can be maintained on the same mobile wallet application.
4. Issuer then verifies the cardholder's identity and mobile device before provisioning the actual payment credentials to the mobile device. Different methods of Identification and Verification of Customer (ID&V) can be used. One typical method is for the Issuer to send an OTP (one time password) to the cardholder's registered mobile phone number.
5. The mobile device makes a request to the Token Service Provider to generate and provision a token.
6. The TSP generates a tPAN (Tokenized PAN) and a tUDK (Tokenized Unique Derived Key) and sends it to the token requestor (TR).
7. The token requestor (TR) generates a set of LUK (Limited Use Keys) using the tPAN and tUDK and sends the tPAN and the LUKs to the mobile device.
   (a) The tPAN eventually replaces the card primary account number (PAN) and expiration date with numeric codes of the same length with a randomly generated token. Separate ranges of numeric codes are allocated so that no payment token can be reversed engineered to find the related PAN number. A mobile wallet usually has five to fifteen stored LUKs on it at one time1.
   (b) Tokens may be generated and downloaded to the device "on-the-fly" when a user is making a payment or else they can be downloaded and stored in a secure location on the mobile device called the Trusted Execution Environment (TEE) and used later during a payment transaction.
   (c) Mapping a token to the original payment card information (PAN) is limited to the TSP or to the issuing bank. The mapping is stored in a TSP token vault.

Checkout

8. The cardholder hovers or taps the mobile NFC device on the merchant's contactless POS terminal.
9. If more than one card of the same type have been enrolled the mobile device will ask the card holder to select which card is to be used for payment.
10. Optionally, the cardholder is asked to authenticate to the mobile device (e.g. through a fingerprint).
11. The mobile device generates the required data, using the previously stored/downloaded tPAN and LUKs, and sends the data to the Merchant's POS. The modes used are either Quick Visa Smart Debit/Credit (qVSDC) mode - in this case the mobile device generates an application cryptogram based on the transaction data provided by the POS, or Mag-Stripe Mode (MSD) mode - in this case the mobile device generates a signature (dynamic CVV) using the Application Transaction Number (ATC).

12. The POS forwards the transaction data to the acquirer, then to the payment network. At this point, the payment network detects the transaction is using a token and asks the TSP to de-tokenize the transaction. The TSP maps tPAN to its PAN and sends the PAN back to the payment network. The payment network forwards the transaction (detokenized) to the card Issuer for authorization.
13. The card issuer generates an authorization response and sends it to the payment network. The payment network asks the TSP to re-tokenize the transaction data. The payment network then forwards the data back to the Merchant's POS. The Merchant completes the transaction based on the authorization response provided.

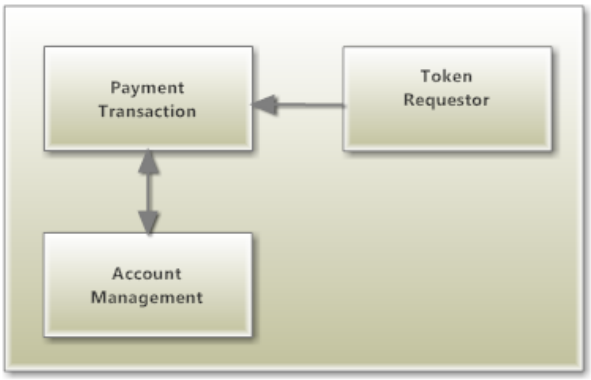
Clearing and settlement

14 -16. During clearing the merchant deposits the transaction purchases information to the acquirer. The acquirer passes the information to the payment network. The payment network verifies the transactions, pays the acquirer who in turn pays the merchant. The payment network passes the transactions to the card issuer, which in turn debits the cardholder's account and pays the payment network. Throughout the transaction process the appropriate interchange fees are deducted by the respective stakeholders.

Defining the transaction as a Finite State Machine

The main contribution of this work is split into two phases. In the first phase, the "task" of the smartphone in the transaction as outlined above, was modeled as a Finite State Machine: this defines a finite set of states and transitions between different states during a payment and during network communication with several online services. In the second phase, the model was studied for security risks with particular emphasis on customer verification and authentication methods, tokenization and the impact in operating cryptographic functions and storing cryptographic keys, required during an HCE payment process. The result is outlined in the next section. The interested reader is referred to the full dissertation for details.

Figure 4: Model Architecture.



Outlined Risks

Identification and verification of the cardholder

The responsibility of ID&V in HCE shifts from the Issuer towards the mobile wallet developer. While in a credit card based transaction, the cardholder could be asked to enter a PIN for ID&V, in HCE, the

wallet application developer can choose from different methods options include PIN or biometric based on the type of device used and the OS. Methods such as fingerprint verification have to be used with caution as, it could be the case that more than one user is enrolled in the phone's template (e.g. a parent adds a child's fingerprint). Moreover, while the issuer has control over which methods can be used, there is no mechanism to enforce priority over the methods. This could lead wallet application developer to downgrade the ID&V method use to a simpler one as this might be more convenient for the cardholder.

Token storage

Through tokenization, the value of the data stored on a mobile phone, is considerably lower when compared to storing the card's PAN and the accompanying cryptographic keys. However, since smartphones are very well connected, they are subject to widespread malware attacks which would have a low impact on the cardholder (stolen token) but a big impact on the issuers who might have thousands of tokens stolen.

To mitigate this risk tokens should be stored in a Trusted Execution Environment (TEE) making it difficult for malware to steal the data. However, within the HCE industry requirements does not mandate tokens to be stored in a TEE and some smartphones do not have access to a hardware TEE.

Token replenishment

Current HCE implementations use one of the LUKs as an authentication to the TSP or Token Requestor, to allow a smartphone to download new tokens. This is a risk, since if a hacker is able to access one of the LUKs stored on a smartphone, the hacker is then able to download further keys, using the LUK. This turns the "limited" use of the key into "unlimited".

Synchronization between smartphone and Token Requestor

The role of the Token Requestor is to provide tokens to the smartphone when these are used up. However, current implementations do not have a means for the TR to ensure that the tokens have been used. This creates a risk if the wallet application requests a replenishment when it has not yet used up the tokens in its store thereby increasing the risk with a large amount of tokens in storage.

Token Portability

Current HCE implementations do not provide a mechanism for a token to be device-bound. Therefore a stolen token can easily be used on another device. This risk could be mitigated through the use of a device fingerprint or a particular hardware ID that is sent to the TR prior to token generation. Using this mechanism, even if a token or a set of tokens are stolen, their use would be limited as the hacker would need access to the device's hardware fingerprint or ID.

EMVCo Assurance Level does not factor hardware & software configuration

The Assurance Level is a property set by the POS as a measure of "confidence" in a transaction. Through this mechanism, issuers could limit risk by requiring a higher assurance level, for example for transactions above certain value. However, current HCE implementations do not have a mechanism to factor the hardware and software configuration of the smartphone used during a transaction. Properties such as the availability of a TEE and the wallet developer could be integrated into the assurance level mechanism to limit the risk from non-reputable hardware and software (e.g. rooted smartphone).

Conclusion

HCE is attractive to consumers because it allows for a quick and convenient way to make a payment. It is faster as cardholder verification can be done on the mobile device using biometrics instead of PINs. It is also more user interactive and the wallet application can provide other services that a physical card cannot, e.g. providing a history of payments. The benefits of HCE is not only limited to cardholders but also to merchants and Issuers.

While convenience is a necessity in today's world, HCE comes with its own set of risks. Thus the industry needs to be more pro-active to ensure that the same level of security provided by a physical credit card is achieved. It is the author's opinion that in order to achieve a secure HCE based payment system, four important requirements have to be met:

- A safe storage for limited use keys and other sensitive data is required on the mobile device. Shifting the data to the cloud does not necessarily eliminate the risk of theft of such keys.
- The generation of tokens (i.e. LUKs) should occur on the mobile device rather than on the cloud. Considering all the risks involved between doing this process in the cloud and on the mobile device, the latter is a safer approach.
- There is a strong need for a method/technique that can be used to uniquely identify a mobile device. The technique should be difficult to tamper with.
- A supply and demand synchronization mechanism between token generation and token usage is required. While tokens are of limited use, a better way of managing them is required.

Biographies

Shana Micallef holds a BSc degree in Business and Computing from the University of Malta. She completed her MSc in Information Security at the ISG, Royal Holloway, and University of London in 2017. She is currently a Senior Officer within the Internal Audit team of APS Bank Malta and is also a volunteer member within the ISACA Malta Chapter. Her research interests are on payment systems and Cyber Security for the financial services industry.

Konstantinos Markantonakis B.Sc. (Lancaster University), M.Sc., MBA, Ph.D. (London) received his BSc (Hons) in Computer Science from Lancaster University in 1995, his MSc in Information Security in 1996, his PhD in 2000 and his MBA in International Management in 2005 from Royal Holloway, University of London. He is the Director of the Information Security Group Smart Card and IoT Security Centre (SCC). His main research interests include smart card security and applications, secure protocol design, key management, embedded system security and trusted execution environments, cyber physical systems, NFC/RFID/HCE security, electronic voting protocols and Internet-of-Things (IoT). He has published extensively in international conferences and journals. He is also a member of the IFIP Working Group 8.8 on Smart Cards. Since June 2014, he is vice chair of IFIP WG 11.2 Pervasive Systems Security. He is an experienced consultant on a variety of topics including smart card security, key management, information security protocols, mobile devices, smart card migration program planning/project management for financial institutions, transport operators and technology integrators.

Series editor: S.- L. Ng