IS HOST CARD EMULATION (HCE) THE BIG ENABLER FOR MOBILE CONTACTLESS PAYMENTS?

An SPA Position paper

December 2015

1. Executive Summary

HCE simplifies NFC implementation by eliminating the requirement of a SE to store mobile payment applications. But HCE also increases the threat for payment credentials to be captured in the mobile device with the subsequent risk of payment fraud.

In this paper, SPA discusses some of the most significant issues related to the security, roll-out and management of payment applications using HCE, and offers recommendations to move forward with a competitive market for mobile contactless payments using both SE and HCE.

This paper does not intend to provide a detailed technical analysis on HCE security.

The following definitions apply in this document:

Mobile Device refers to mobile phones and smartphones equipped with an NFC controller and host payment applications using either one or more SE, or HCE functionality, or both.

We refer to Android OS mobile devices, because HCE is the NFC functionality of the Android OS.

Secure Element (SE) refers to a chip emulating a card in a mobile device and accessed using the NFC mobile controller.

The Secure Element is isolated from the mobile operating system and hardware, and therefore provides the security features of a certified smart card to a mobile device: secure storage, an isolated and secure execution environment, and hardware-based cryptography. The SE also stores cryptographic keys and execute protocols for the remote management of the mobile payment application.

Card Emulation (HCE) refers to a software module embedded in a mobile device emulating a card and accessed using the NFC mobile controller.

The HCE is not a secure environment, meaning that other applications resident in the mobile device, malicious or not, may compromise the integrity of payment applications. To mitigate this risk, specific security mechanisms are required. They are discussed hereafter.

2. Introduction

An electronic payment transaction is the result of the generation, transmission and verification of a pre-defined series of messages – each conveying specific sensitive data that needs to be protected.

Naturally, the security and effectiveness of such transactions have benefited significantly from the IT revolution of the past three decades and the continuous optimization of the algorithms that represent, encode, store, access and transmit digital data. Similarly, developments in the security of computing devices, telecommunication networks and database facilities have likewise had a positive impact on the processing of this sensitive payment data – with smartphones increasingly the access devices to a vast array of mobile financial services.

It is assumed that to succeed, a new payment instrument must be easy to use, low cost, accepted everywhere, be trusted by the user and backed by a sustainable business model. Trust here is key, and requires that the chosen payment instrument not only be secure, but be perceived as being secure. So that in the event of fraudulent transaction the consumer is certain to be protected and refunded.

The above conditions require that any newly-marketed payment instrument is also interoperable; being compliant with existing payment and communication standards that specify a protocol stack to be implemented. This constraint is observed even in payment products issued by Apple. While the company may not have a reputation as a standard promoter, or for designing products following accepted technical standards, ApplePay does feature NFC-compliance and supports EMV applications.

This paper however focuses on Host Card Emulation (HCE), a technology that offers an excellent example of the challenge of accommodating business, functional and security requirements in a single payment product; that must (1) feature compatibility with existing terminals and (2) manage the coexistence with Secure Elements without undermining the almost-zero level of fraud achieved with payment cards.

3. Level of adoption and security of different contactless payments

Mobile payments are the natural evolution of traditional payment cards. Yet, deploying mobile payment solutions creates extra complexity because (1) the payment value chain is expanded and (2) the issuer of the payment application, usually a financial institution, does not own the hardware where its application is installed. The issuer can, of course, operate directly or can outsource the infrastructure required to manage its mobile payment applications to a third party. And in an operating environment where the regulatory framework is driving down the profitability of traditional card payments, financial institutions have expressed an obvious interest in those mobile technologies they might keep under their complete control.

HCE eliminates the cost, time and energy required to negotiate the price conditions to use the Secure Element owned by a third party, to host banking applications. Unsurprisingly therefore, HCE is being perceived as a key mobile technology to drive mobile payments operated by financial institutions.

HCE addresses the issuers' desire to deliver their own contactless mobile payment services to their customer base. Institutions benefit from reduced time-to-market and through a target service cost that can be unilaterally fixed, without negotiating with telecom operators. Little wonder then that many HCE pilots are under way both in the European Economic Area (EEA) and worldwide.

HCE also supports the adoption of mobile contactless payment terminals by retailers – thanks to the very large penetration of Android smartphones in consumer markets. Likewise, adoption rates will grow should the mobile device be increasingly perceived as a secure payment instrument.

Users are already comfortable with using contactless cards in tap-and-go mode. And despite very high adoption rates of contactless payment cards, we have seen no real increase in levels of fraudulent activity. Instances of fraud on contactless cards are in fact extremely rare, with losses of less than a penny for every £100 spent on contactless – far lower even than overall card fraud. (UK Cards Association, July 2015). The result is that contactless payment cards are trusted by the consumer.

However, compared with contactless cards, securing HCE implementations faces two challenges – to mitigate the well-known Android OS vulnerabilities, and to avoid downloading malware over the different channels offered by the rich connectivity capabilities of the mobile device.

From the security engineering prospective, the most obvious threat mitigation countermeasure is the isolation of mobile payment applications from the vulnerable components of the mobile device. But that is the job of the Secure Element, not HCE.

However, SPA is in no doubt that the intrinsic security of HCE applications is important. HCE security evaluation must be mandatory for all (currently it is optional for issuers) products, and requires some form of standardization. Today there are different rules applicable for HCE issuers, with an optional certification requirement. In this respect EMVCo has announced the publication of Security Requirements for HCE implementations in Q1 2016.

4. The Mobile Payments security conundrum

The latest generation of smartphone operating systems are as complex as those found in the desktop environment. Isolation and sandboxing provided by mobile OS is regularly broken, and consumers often root their device and in so doing risk sensitive data leakage. The commercial success of Android OS mobiles makes them key targets for hackers, and the latest Black-Hat 2015 event focused on new threats on Android OS, including attacks that do not appear to be purely theoretical academic exercises.

In this context, HCE represents an unprecedented step forward in the risk appetite of payment application issuers, as it drops the fundamental security paradigm: that a sensitive application must only be run in a secure computing platform, certified as tamper resistant. At present mobile devices using Android OS do not feature such security properties. However, the growing volume and severity of potential threats will drive improved security countermeasures offered by Android OS to protect HCE applications. Google has committed to doing so, and we are already seeing significant R&D investment in securing software implementations.

The lack of a hardware SE for payment credentials and data in HCE means that protection must come from a combination of specific software-based security techniques, either cryptographic or not, including code obfuscation, white-box cryptography, use of payment tokens, active threat monitoring in the mobile device and enhanced risk scoring process in the issuer side prior to the transaction authorization.

It is SPA's view that while such countermeasures remain unproven, they are certainly being highly scrutinized by security experts – many of whom offer differing opinions on the actual level of protection they provide. SPA believes it is important to point out that unproven does not mean unsafe, and that international payment networks are granting certificates for HCE implementations compliant with their proprietary specifications.

The security policy for HCE, and therefore the level of risk for the payment application, will differ depending on the risk appetite of the payment application issuer. This approach is dangerous and could lead to a major security incident. SPA considers that to achieve a level playing field, and to properly protect the consumer, an in-depth security industry discussion is needed to assess HCE vulnerabilities and specify a minimum set of security requirements for HCE implementations.

As previously discussed, specific software countermeasures can have an impact and should be implemented as these reflect state-of-the-art protection as we see it today. However, the feeling in the security industry is that these countermeasures are challenged as new weaknesses of Android OS are uncovered. As such, HCE security discussions should consider tokenization or alternate Primary Account Number (PAN)s to mitigate security threats.

While it is true that tokenization and HCE are technologies addressing different concerns, they complement one another very well. Indeed, today many security experts consider tokenization to be critical to the success and future adoption of HCE. There's no doubt HCE presents security vulnerabilities, and payment tokens may increase the level of security assurance of such implementations. In this context, the EMV tokenization framework is the most visible initiative to secure card payments, but SPA would point out that other technical options are also possible. The use of an alternate PAN with dedicated Issuer Identification Number (IIN) range provides a real alternative to tokenization implementation.

Of course, the possibility of a competitive market for payment tokens is much debated. The generation and use of payment tokens (other than EMV) that retain backward compatibility with card processing systems is technically feasible. For instance, the GSMA has started work on the specification of a SIM-based tokenization service, enabling bank issuers to reach customers through the mobile telecom operator. However, it appears that financial institutions favor a single tokenization system, using EMV specifications for global interoperability, and as the basis for competition between Token Service Providers. In this respect PCI is in the process of specifying the security requirements for EMV Token Service Providers in an effort to protect cardholder data through the tokenization process.

5. HCE Vulnerabilities and how EMVCo Tokenization may mitigate security risks

In-field interoperable products can only be designed using stable standards and/or implementation specifications. Here the Android KitKat 4.4 HCE specification appears functionally stable whilst the EMV tokenization framework is still something of a moving target.

One of the reasons for this delay is that the initial EMV tokenization framework specification creates problems for some large retailers that are using the PAN for customer account management purposes. EMVCo is well aware of the problem and the integration of the Payment Account Register (PAR) data element in the next release of the EMV Tokenization Framework should solve the problem - opening the door to the acceptance of HCE/EMV Token-based mobile payments.

Most of the recent m-payment solutions including Apple Pay, Android Pay, Samsung Pay and even wearables use Near Field Communication (NFC) technology and EMV tokenization. Android Pay aside, the other products use an embedded SE to protect payment applications. In theory, the combination of both the SE and a token is considered to be the most secure combination available in the mobile environment today.

Of course, actual security is more complex: the product must be properly designed, implemented and thoroughly tested. And the service must include a robust enrolment customer process based on realistic assessments of what might go wrong. Indeed, the initial and unexpected high fraud rates of ApplePay where due to security holes during the enrolment process.

From the security engineering prospective, Android OS memory is also a vulnerable environment. Attackers could gain access to payment credentials when stored in the HCE to be used later, or install malware applications to take control of the mobile OS to make fraudulent payments. In August 2015, Karpersky Lab published its cyber threat report detailing a huge volume of identified threats in Q2. It found around 291,800 new mobile malware programs targeting financial applications had emerged during 2015 Q2 – a 2.8 times increase on Q1. The Q1 Kaspersky Lab report mentioned Trojan-SMS.AndroidOS.OpFake.cc, which was capable of attacking dozens of mobile banking and financial applications, and a new, more powerful version of the Trojan ready to emerge by the end of the year.

It is therefore vital that security policy for HCE urgently adapts as the threat environment evolves.

One way to reduce risks is by locally storing tokens that are valid for a single EMV transaction. The token adds security because it cannot be used beyond its pre-defined purpose, for instance to pay a particular retailer. This makes token-compromise a much less attractive option for the attacker than, for instance, capturing the PAN, expiry date and the CVC of the card – which will be accepted everywhere.

So should a NFC/HCE solution use EMV payment tokens? To answer this question, it is important to

(1) analyze the impact of HCE/tokenization in existing systems and assess if the operational advantages of HCE are somehow undermined by the integration of a token management system,

(2) determine the HCE security risks mitigated by the use of EMV tokens and

(3) evaluate the residual level of risk.

5.1. The impact of HCE /tokenization integration on existing systems is twofold:

- The token must be generated from a PAN; provisioned; and then when used to pay, de-tokenized prior to the authorization request. This process requires the intermediation by a Token Service Provider (TSP) - the only entity in the system generating tokens and able to detokenize them. The integration in the card system of the TSP makes the payment processing architecture more complex, adding cost – although it is also the case that many banking members of international schemes are already investing in the integration of TSP systems into their card payment processing infrastructures.
- The impact at the mobile device level, will depend on the design of the application manager:
  - the token provisioning mechanism in the HCE

- the way to manage the coexistence between HCE and SE applications. This point is briefly discussed below.

To minimize the impact on existing card payment systems, two models mimicking a mobile contactless payment are currently deployed for HCE payments:

(1) In the standard EMV model, upon customer enrollment, the issuer instructs the TSP to provision the HCE with a static token and a number of session keys. At every tokenized transaction, the HCE uses one/more session keys assigned with the token. When no more session keys are available a mechanism exists for the TSP to generate and provision the HCE with more session keys, without necessary replacing the original token. In this model, the TSP can be positioned between the issuer and the acquirer in the card processing chain, or can be directly integrated in the issuer processing facilities.

(2) When using an alternate PAN, the payment application issuer stores the alternate PAN and associated keying material in the HCE. The HCE protects these sensitive data using obfuscation technologies. When paying, the HCE uses the alternate PAN and keys to run a conventional EMV contactless payment. No TSP is required and the authorization request message is generated by the acquirer using the alternate PAN and associated EMV cryptogram.

In addition to existing solutions, advanced architectures using cloud services to protect payment credentials and/or generate and download tokens in the mobile have been discussed. No commercial offers are available at present for TSP and HCE in the Cloud. However, they certainly will be in the future. Indeed, SPA has presented a taxonomy of these advanced cloud architectures for HCE, and the pro's and con's, to the Innovation Expert Team of the European Payments Council - Card Stakeholders Group (EPC-CSG)[1].

The security risks mitigated by the use of tokens for HCE are related to cross-channel fraud. An HCE implementation without tokens involves long term storage of the PAN and associated keys in a vulnerable environment. If stolen, a fake HCE with this data can be created and used to pay in any retailer accepting contactless transactions. The use of an alternate PAN for HCE bound exclusively to NFC contactless payments prevents cross-channel fraud. Likewise, tokens offer stronger defense against cross-channel fraud because they are issued for a specific channel and a specific retailer.

The transaction data in the cloud is secure at rest, provided that access to the cloud is effectively controlled. Likewise, data is secure in transit if a secure channel is established between the cloud and the HCE. The term "secure cloud" refers to compliance with Appendix D of the PCI-DSS Cloud Computing Guidelines. It is also important to say that access to cloud services raises other security concerns related to key management that are out of the scope of this document. The security architecture of distributed systems is complex, meaning many heterogeneous technologies must coexist without security gaps. Due to this complexity it is difficult to cover all the risk scenarios. In the HCE case the problem is more challenging because of the intrinsic vulnerability of the Android OS kernel. However, more complex does not mean unfeasible. It simply means more interfaces and more processing; increasing the number of attack points in the system.

[1] *The EPC-CSG is organized in 5 sectors including Banks, Schemes, Processors, Retailers and Vendors. In all 27 members:*
☐ *25 members from the 5 sectors including the chair (EPC), the co-chair and*
☐ *2 observers: ECB (European Central Bank) and the EC (European Community) DG Market.*

The EPC-CSG mission, organization and governance are defined by its Terms of Reference. The EPC-CSG scope includes any payment card product regardless the form factor. Thus, the EPC- CSG has taken over the maintenance of the Volume Book of Requirements, the core specification for SEPA for Cards.

Tokenization is likely to be adopted for those payment application issuers who choose to go down the HCE road. The proper use of tokens significantly improves user security for any kind of card payment. In the HCE scenario, tokenization can be considered as a real enabler. Yet the EMVCo tokenization framework has received some criticism from its ambiguity and lack of strong recommendations in terms of implementation. However, in the same way the EMV specifications have been upgraded to cope with new attacks regularly reported, the same continuous improvement process will take place with subsequent releases of the EMVCo tokenization framework.

6. Moving ahead: Improving the coexistence of HCE with the SE

Payment schemes and issuers look to be willing to accept a lower security level for the initial NFC/HCE implementations in an effort to boost the number of NFC mobile payment transactions. However, should an increase in fraud rates be observed, many payment application issuers will choose to put their data into a SE, should one be available on the mobile device. In this scenario, different mobile payment applications could reside in both the HCE and the SE. They would coexist and the mobile device would need a mechanism to make each visible and selectable. Global Platform (GP) is currently specifying this mechanism, called the Managing Entity, to enable the user to activate/deactivate mobile payment contactless applications regardless where they are stored – in the SE or HCE.

The GP initiative will therefore enable a hybrid solution, which appears to be well aligned to the European Regulatory framework. This stresses the consumer’s right to select the payment application of choice.

Ultimately though, For HCE, freedom of choice will depend on Google’s willingness to use the SE on Android devices.

7. Standardizing requirements for HCE implementations: some questions to answer

The innovation the market offers for mobile contactless payments has been recognized by the CSG. The last draft of the CSG Volume Book 4 points out that (1) transactions may be initiated using HCE and that (2) “future releases of the Volume may cover those as the technologies further mature”. The Volume therefore ignites competition between the SE and the SE software emulation (HCE). A level playing field is important. Competition should be open, secure and responsive to the needs of the users.

From the retailer perspective, the contactless terminal must ensure an application resident in the HCE can be selected in the same way than an application resident in a SE. From the consumer perspective, the ability to choose or pre-select the level of priority for the resident mobile payment applications must not be impaired. For liability shift purposes, the security risks for the different stakeholders must be understood and controlled. A first step is the addition in the Volume (SEPA Book of Requirements) of a minimum set of security requirements for HCE implementations, after an indepth analysis of vulnerabilities and risks.

8. Conclusions

(1) To minimize security threats in the payment card value chain, the different security controls should fit together in a consistent security policy for the operators of a mobile payment system. This does not mean that all these controls provide the same levels of security.

(2) SPA considers that for HCE to become a viable alternative to the SE, the minimum security requirements for HCE implementations should be harmonized through a standardization process. These requirements should include countermeasures (1) against software modification threats, (2) against software threats intended to expose sensitive data or (3) any other threat that makes the HCE behave in an unintended or unauthorized way.

(3) Combining payment tokenization with techniques to encrypt and hide security keys and sensitive data in the code of mobile payment applications is a good solution to secure HCE. Yet storage in the HCE of an alternate PAN may be considered as well. Both solutions are enablers for HCE, which in turn is an enabler for mobile contactless payments.

(4) The EMVCo Tokenization Framework provides the interoperability of tokenization implementations proposed by international networks. Yet other payment tokenization schemes could fulfill the same security objectives for HCE implementations. Alongside competition between SEs and HCE for managing payment applications, a second level of competition could also take place between payment tokenization solutions.

(5) In order to enlarge the offer for mobile payment solutions and ensure consumers are able to choose their preferred application it will be important that mechanisms for the coexistence between payment applications stored either in the SE or in the HCE are developed and supported in future versions of Android OS.