Organisational Aspects and Anatomy of an Attack on NFC/HCE Mobile Payment Systems

Maurizio Cavallari[1], Luca Adami[1] and Francesco Tornieri[2]
[1] S.E.Gest.A., Università Cattolica del Sacro Cuore, via Necchi 5, Milano (MI), Italy
[2]BKG Laboratories, Verona (VR), Italy

Keywords: Organisation, Security, Mobile Payments, HCE, NFC, RFID.

Abstract: Near Field Communication (NFC) and contactless applications are increasing at unprecedented rate and their value is being recognised by the financial industry (Ok et al., 2011). Attacks are also increasing and they can compromise the business value on NFC applications (Murdoch and Anderson, 2010, Trend Micro, 2015). The present paper analyse the anatomy of possible attacks, uncovering vulnerabilities and suggesting possible countermeasures. The value of the paper is found in the contribution to practical mitigation of risk in the mobile payment financial business, with respect to the technology side. Host Card Emulation (HCE) is a technology solution that permits the creation of a virtual representation of a smart card using only software components, effectively eliminating the need for Secure Element hardware in the device. NFC/HCE technologies has proved itself very vulnerable in a variety of aspects. The paper would go through specific vulnerabilities and vulnerable situation, like: a non-secure-device/cloud communication channel; access to data saved locally in wallet; reusability of token; use of fake POS; malware and fake application; specific vulnerabilities of "Tap & Pay"; device/cloud decoupling. Countermeasures that have been proved effective are offered to readers along with Organisational aspects to be taken into account.

1 INTRODUCTION

Host Card Emulation (HCE) is a technology solution that permits the creation of a virtual representation of a smart card using only software components, effectively eliminating the need for Secure Element hardware in the device (Smart Card Alliance, 2014).

Before going into detail about the concepts characterizing HCE, a definition of the technology at the basis of this solution is opportune: Near Field Communication (NFC).

NFC is a technology that provides connectivity - unlike its predecessors such as the contactless smart card - two-way and short-range: when two NFC devices (the initiator and the target) are matched (within a radius of about 4cm), a peer-to-peer connection between the two is created and both can send and receive information (ISO/IEC 14443 A&B, 2011; JIS-X 6319-4, 2005).

NFC can be achieved directly via a chip integrated in the device or through the use of a special external adapter that exploits the ports of the SD card or phone micro SD card. It is also possible to make an NFC-enabled device communicate with a passive NFC chip, called "tags".

The main cases for using this communication mechanism are mainly in the financial world: NFC-enabled devices can be used in contactless payment systems in the mobile environment. Google Wallet, for example, allows users to store credit and loyalty cards in a virtual wallet, and then use an NFC-enabled device to make payments to terminals that accept transactions channelled through the MasterCard PayPass circuit.

The basic components necessary to make an NFC transaction are (Halgaonkar et al., 2013):

- NFC Controller: An electronic component that resides within the mobile device, by which it is possible to communicate with another NFC device (for example, a POS, i.e. Point Of Sale, equipped to read contactless cards);
- Secure Element (S.E.): A secure physical space that can be housed inside the SIM or the device itself, rather than in removable hardware (micro-SD Card), which hosts a payment application called MCPA (Mobile Contactless Payment Application). MCPA is an EuroPay, MasterCard,

Cavallari M., Adami L. and Tornieri F..
Organisational Aspects and Anatomy of an Attack on NFC/HCE Mobile Payment Systems.
DOI: 10.5220/0005477506850700
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (WOSIS-2015), pages 685-700
ISBN: 978-989-758-097-0

Visa (EMV) compliant payment application, very similar to the one installed in the microchip on the majority of credit/debit cards. This is the element in charge of emulating a payment card;

- Trusted Service Manager (TSM): The external technology that deals with the provisioning of the MCPA and its entire life cycle (and which therefore has access to the Secure Element).

In this way, a payment transaction occurs in Card Present mode, and the values of the Merchant Service Charge (MSC), i.e. the amount the Acquirer charges the merchant, are normally lower than those for Card Not Present type transactions (Issovits and Hutter, 2011).

Note that in this approach, a smartphone NFC chip is authorized to communicate only with applications (MCPA) installed within the Secure Element. A representation of the typical flow of a transaction is shown in Figure 1: the card to be emulated is made available by the Secure Element, to which the NFC controller directs the data received by the NFC reader (Patidar et al., 2011).

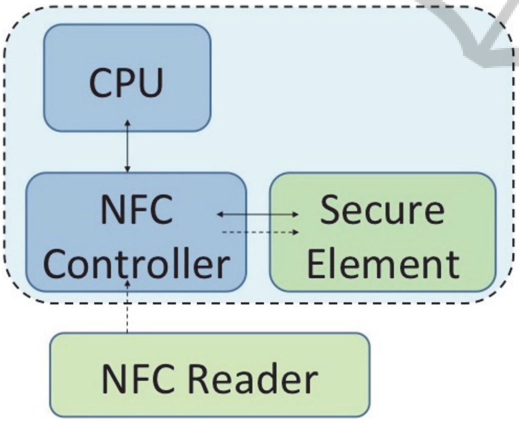


Figure 1: NFC Function.

NFC is widely used on terminals equipped with version 4.0 (Ice Cream Sandwich), such as Google Nexus and some devices manufactured by Samsung. On the release of Android 4.4 (KitKat), Google subsequently introduced support for secure transactions based on NFC via Host Card Emulation, as presented later in this chapter. In systems based on iOS (Apple) the NFC chip is pre-installed in Iphone 6 and Ipad Air 2 (the chip was missing in earlier devices). The current market of mobile payment systems is based on solutions designed for Android OS, and marginally, due to diffusion, for Windows Phone systems.

The Host Card Emulation solution permits the creation of an exact, virtual representation of a smart card using only software components, effectively eliminating the need for Secure Element hardware in the device (Hancke, 2007). In particular, HCE is defined as the ability to exchange information via NFC between a terminal (enabled for the exchange of information with an NFC card) and a mobile application configured to emulate the functionality of an NFC card (Devendran et al., 2012). The novelty lies in the fact that HCE enables the NFC protocol to be routed directly through the operating system of the device rather than being first processed by Secure Element hardware, such as a chip configured to act simply as a card, without other possibilities. A representation of the innovation introduced by HCE is shown in Figure 2.

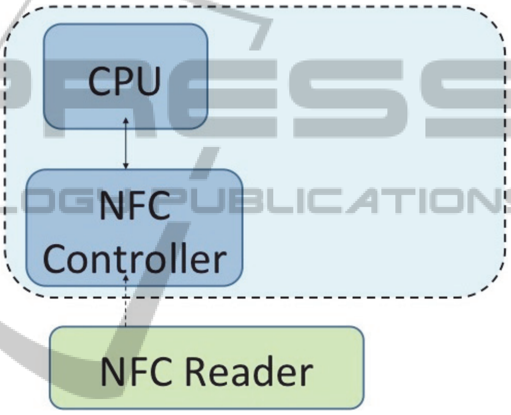


Figure 2: HCE: elimination of the Secure Element.

HCE can therefore be defined as an open architecture on which it is possible to develop application solutions that have the possibility of emulating the payment card and that interface directly with the NFC Controller, without the need to resort to Secure Element hardware available on the device (Worstall, 2012). When a smartphone is placed near the reader, the NFC Controller calls an Android (HCE service) previously associated with identifiers reserved for payment networks that identify payment applications (at present no information regarding the workings of the payment solution for Apple products is available, as these not yet on the market).

Having eliminated the need for Secure Element hardware in the smartphone, card data to be emulated is stored by the issuer authorisation centre in the “cloud”: the payment application’s task is to recover the information needed to make the payment from the cloud. This shift, however, is not to be considered only as a modification of the data flow (data movement) but also as the renouncing of the use of the secure hardware cryptographic capability of the Secure Element, while storing and processing such data in a secure session with the terminal: the mobile device becomes a transducer that connects the POS terminal with the image of the card stored in the network (Verdult and Ois Kooman, 2011).

The current process between POS and Secure Element is divided into two parts by delegating the task (local) of the Secure Element to applications on issuer card images (remote).

This solution also simplifies the chain, reducing the role of third parties and enabling freedom from mobile network operators and, contrary to what happens with the classic NFC, from any kind of conditioning by SIM or SD card, with a consequent reduction in fees. At the same time, the current network consisting of POS terminals, requires no changes for using contactless cards or NFC-enabled smartphones with Secure Element hardware.

Further advantages of the HCE solution also include the simplification of the provisioning process and a virtually unlimited storage capacity. Access to provisioning data is possible online through the use of simple login credentials to the secure host. At the same time, the use of HCE entails no theoretical limit to the storage of card data as it is saved in the network (whereas the reserved space in Secure Element hardware is limited by the physical characteristics of the individual device).

Note that HCE payments can also be made even where there is no cellular network. Whilst the dislocation to the cloud of the Secure Element implies the need for the terminal to access certain data (e.g. payment token) hosted on the network before being able to make the payment, in order for the outcome of the payment to be unaffected by the lack of a network connection, retrieval and renewal of payment data occurs in the background, on a continuous basis which is transparent to the user. In this way, a mobile network connection is not essential during the payment transaction (Haselsteiner and Breitfuß, 2006; Jules and Weis, 2005a).

1.1 HCE Infrastructure

The HCE architecture in Android (shown in Figure 3) revolves around the concept of "HCE services"), that run in the background without any user interface. This is convenient since many HCE applications (like loyalty cards) do not require the user to launch any application to use the service, but simply place the device near to an NFC reader to start the service correctly and execute the transaction in the background (Smith-Strickland, 2013).

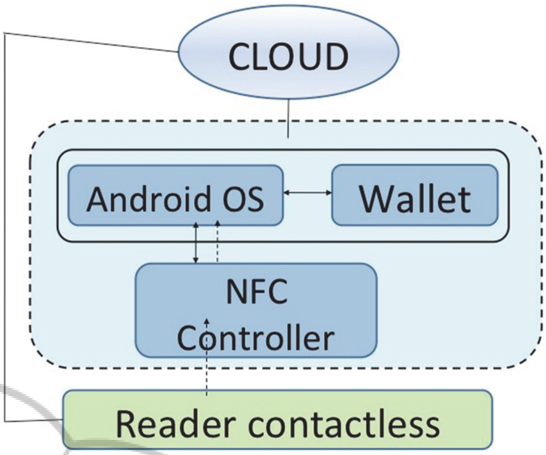


Figure 3: High-level architecture.

When the user places the smartphone near to an NFC reader, the Android operating system must be able to identify which HCE service is going to be contacted. To solve this problem, and to define a means of selecting the wallet application required, the concept of *Application ID* (AID) is used. These AIDs are nothing more than a sequence of 16 bytes that are well known and publicly recorded by the major payment networks (such as Visa and MasterCard), and used by Android to determine which service the NFC reader is trying to contact.

Mastercard and Visa use the following AIDs:

- `MasterCard "A0000000041010"`
- `Visa "A0000000031010"`

In particular, note that the Android implementation of HCE was created to allow simultaneous support to other card emulation methods, including the use of Secure Element hardware (the "standard" NFC). This coexistence is based on the principle of "AID routing", according to which the NFC controller maintains a routing table consisting of a list of rules, each of which containing an AID and a destination (which may be either the CPU running Android applications or a Secure Element). The first message sent by the NFC terminal contains the `"SELECT AID"` command, which is intercepted by the NFC controller, which then controls if the specified AID belongs to at least one of its routing rules. If there is a match, the destination in the routing rule selected is identified as the recipient of the subsequent NFC communication.

The main difference between the two routes then resides in the recipients of the communications. If the target is the Secure Element everything proceeds in the traditional method of contactless payments via NFC. If communication is HCE, the controller routes the communication directly to the operating system. Android considers as payment applications all those HCE services that have declared, in their manifest, the use of AIDs in the payment category. Furthermore, Android contains a “Tap & Pay” menu in its settings, which lists all these payment applications. In this menu, the user can select the default payment application to be used when the smartphone comes into contact with a payment terminal (POS).

1.2 HCE Support

Currently, payment solutions based on HCE are supported by all those terminals enabled with NFC-connections and which use Android 4.4 (the benchmark platform for the entire industry) as the operating system, given the spread of Android at the expense of Windows Phone (NB. the NFC chip is only a recent addition to Apple devices, iPhone 6).

In its implementation of HCE, Google (now owner of the system) states that Android 4.4 supports many protocols and standards adopted by the most common types of NFC readers, for which contactless payment applications have currently been developed. In particular, the new operating system implements card emulation based on the NFC-Forum ISO-DEP specifications (in turn based on ISO/IEC 14443-4) and is able to handle the Application Protocol Data Units (APDUs) defined by ISO/IEC 7816-4.

2 LITERATURE REVIEW

Near field communication has been leading important changes due to the methods in which consumers purchase goods and services, and this attracted a number of scholars and researchers to investigate further on security aspects (Ozdenizci et al., 2010). Slade, Mayes among other scholars, researched into consumer adoption of proximity mobile payments and NFC many different potential commercial applications, ranging from marketing to nutrition, transportation, gaming, and health care (Slade et al., 2014; Mayes et al., 2009).

Using proximity payments, it is possible to create a new link between physical materials and digital information (Juels and Weis, 2005b). Proximity payments are possible because, during the last decade, industry had been developing the new technology, Host Card Emulation (HCE), which allow to create an exact virtual representation of various electronic cards, eliminating physical supports (Ok et al., 2011; Van Damme et al., 2009).

In one of the last work of Paya, *“HCE vs embedded secure element: relay attacks”*, he discussed a series of potential vulnerabilities on HCE, that allows possible intrusion by malicious attacker, related to outdated software and low security levels (Paya, 2014). Other studies show similar results, suggesting the relay attack is an actual breach of NFC/HCE (Van Dullink and Westein, 2013; Ozdenizci, 2010).

As a consequence the industry related association “Smart Card Alliance” produced a white paper “Host Card Emulation (HCE) 101” in which they explained all possible implications concerned payment application and security (Smart Card Alliance, 2014, Honig, 2013, Juels et al., 2005).

The present paper is focused on the anatomy of possible attacks, starting from a series of contribution on NFC payments attacks, matching more practically studies like Van Damme et al. and Madlmayr et al. (Van Damme et al., 2009; Madlmayr et al., 2008; Aigner et al., 2007) about NFC mechanism and functioning, technical exploration such as exposed by Ok et al, in which we get a sense of real magnitude of NFC applications (Ok et al., 2011; see also Suman, 2013).

The present research work is going through specific analysis of vulnerabilities and vulnerable situation, updating past research findings as Van Dullink and Westein, where they research into attacks using NFC enabled devices (Van Dullink and Westein, 2013). Furthermore, is possible to identify a range of possible implications on mobile payment security that we describe based on past research findings by Worstall, Nai-Wai and Li, Li et al. (Worstall, 2012; Nai-Wai and Li, 2012; Li et al., 2014).

Organisational aspects and literature about behaviour, technology human-interaction, learning and trust, are to be taken into account as the organisation of attacks relay on human behaviour in the first place and on technology and application functioning, at second (Avison and Wood-Harper, 2003; Straub et al., 2008; Za et al., 2015).

3 ORGANISATIONAL ASPECTS OF NFC/HCE SECURITY

Within academic literature we can observe a lack of specific studies directly pointing to the relationship between trust and flexibility and security and attack on technology behavioural models (Hagen et al., 2008, Cavallari, 2008, Juels and Wies, 2005a). The novelty of the present paper is to link together the established organisation science literature trust and flexibility along with the extant literature on information systems security (Cavallari, 2011; Marzo and Castelfranchi, 2013).

Notable and renowned theories and authors' conceptualisations about flexibility and information systems security are regarded as the start for further speculation (Avison and Wood-Harper, 2003). The innovation of attack on NFC/HCE described in the paper could have been identified with respect to specific conceptual domains are then utilised to build up a coherent organisational theoretical framework (Straub et al., 2008).

In order to explore the flexibility of an organisation we addressed the research regarding organisation's flexibility looking at the contextualist approach by Pettigrew, which is relating flexibility to organisational matters (Pettigrew, 1987, 2001). Major contributions to the proposed theoretical framework are coming from Burgelmann and Gupta so to describe the baseline organisational conditions in order to make effective the explained attack on NFC/HCE (Burgelmann, 2002, Gupta, 2006).

Information security and specifically the anatomy of attack described in the present work and the organizational theories are in particular link and shall not be regarded as two aspects of the patterns to identify the attack, but as a whole behavioral path which lead to secure (and possibly insecure, if the whole picture is disregarded) transactions on mobile payments.

The most appropriate theoretical organisation science framework we found adequate is a *contextualist* approach within change management, proposed by Pettigrew and commented by recent research. The major contribution is certainly the intent to create *"theoretically sound and practically useful research on change",* that explores the *"contexts, content, and processes of change together with their interconnectedness through time (cit.)"* (Pettigrew, 1987, 2001; Cavallari, 2011).

Research studies on organisation theory have proven that the new types of attack on technology are benefited by, and can be researched into the exploitation and exploration learning theory (March, 1991, Gupta et al., 2006). Exploitation learning comprises refinement, choice, production efficiency, selection, implementation and execution; whilst exploration learning includes search, variation, risk taking, experimentation, play, flexibility, discovery and innovation (Burgelmann, 2002; Spagnoletti and Resca, 2008).

Empirical studies conducted by He and Wong demonstrate that exploitation learning leads to the development of operational capabilities, in the first stage (He and Wong, 2004). In the second stage of an exploration learning organization a more participatory leadership style fosters experimentation and risk taking (Straub, 2008). The mentioned authors highlight that, within the boundaries of organisation theory and information systems security, exploration learning facilitates flexibility permitting development of new processes, and therefore new attacks on technology and information systems – such as the mobile payments infrastructure (He and Wong, 2004, Straub, 2008). The basic distinction between organisational learning and organisation adaptation is well discussed by Fiol and Lyles and refined further by Avison and Wood-Harper, as they show that change do not necessarily imply learning (Fiol and Lyles, 1985; Avison and Wood-Harper, 2003).

Trust and dependence network has proven to be a major organisational aspect to be taken into account when dealing with learning, knowledge sharing and innovation. This particular concept, i.e. trust, is directly translated into human actions while dealing with technologies, for instance with payment applications provided by banks or issuers, and also translated between applications/devices interaction (Za et al., 2015).

Other organisation theory authors like Benner and Tushmann, base their discussion on the propositions of March and Levitt and then argue that while process management activities are beneficial for organizations in stable contexts, they are inconsistent with incremental innovation and change (Benner and Tushmann, 2003). So with respect to the latter organisation research findings we shall assume the identification of innovative and new attacks shall come not from operational processes, but rather on "out of the box" and parallel thinking (March, 1991, Levitt and March, 1988).

4 SECURITY ASPECTS OF HCE ATTACK

HCE provides that communications use the Android operating system as a vector. This ensures the use of some, albeit basic, security controls, such as the use of sandboxes that prevent an application from accessing data from another application. However, these guarantees are void if the device is tampered with (known as "rooting" in Android/Windows Phone systems or "jailbreak" in Apple systems).

Note that there are various scenarios that can lead to the root of a device, such as:

1. The user carries out the procedures necessary to obtain root permissions on his phone;
2. Malware capable of tampering with the device, taking advantage of vulnerabilities in the operating system itself or in some drivers. In this case the difficulty of introducing into the operating system the necessary countermeasures to mitigate those vulnerabilities must be considered;
3. If lost or stolen, the device could be tampered with to gain access.

A phone that has been tampered with no longer makes security checks on transactions made by the user, who will have total control of the phone and full access to all memory partitions (and also those of the system). A potential attacker would be able to access all the critical information contained in the phone, such as payment credentials that could be used to conduct fraudulent transactions. Furthermore, a malicious tampering application installed on a mobile phone could gain full control of communications (and carry out a *Man-In-The-Middle* type of attack), and even control and interact with other applications.

Finally, since the routing table of the NFC controller can be changed by the operating system, a malevolent application could cause *Denial-of-Service* and silently go about changing all the NFC service routes on the phone (normally, this would require the explicit consent of the user).

The following analysis considers the hypothesis that the device has not been tampered with and also presents the principal vectors of attack; in a number of cases the effective vulnerability with be explored.

4.1 Non Secure Device. Cloud Communication Channel

The communication channel between smartphone and the cloud platform that hosts payment services is a critical factor in the infrastructure of the HCE solution. Above all, it is used in the two critical activities of *Enrolment* and *Account Provisioning*.

During the Enrolment phase, the user inserts the card data, or selects one of those available in the digital wallet, to request the service (McHugh and Yarmey, 2012). This data, which is then forwarded to the cloud platform, includes confidential details as:

- Name and place of birth of the cardholder;
- PAN (Primary Account Number);
- CVV2;
- Expiration date.

In some cases, the user may also be required to choose an access code (called *Consumer Device Cardholder Verification Method, CDCVM*), which will in turn be forwarded to the cloud platform and which will thereafter have to be entered directly in the wallet application before making a payment, in order to verify its authenticity.

After Enrolment, the system creates a binding between application, smartphone and CDCVM and is able to begin the process of Account Provisioning, during which so-called "*payment tokens*" (virtual PAN generated from an access PIN for the HCE service) are generated. These tokens are then sent to the smartphone to be stored and enable the user to make transactions even when no mobile connection is available.

If the channel connecting the smartphone with the cloud service is not secure, an attacker could intercept the data of the original card (such as PAN, CVV2, and expiration date), for card cloning, and even as the fundamental fields for the HCE service, such as the CDVM and payment tokens. A possible scenario: an attacker could install an untrusted CA certificate on Android and so create a "MitM" (Man in the Middle) attack (Momani and Hudaib, 2014, Patidar and Bhardwaj, 2011). This situation could be summarize in these steps:

1. An attacker creates a fake gateway
2. The fake gateway intercepts the TCP traffics and sends to the device an untrusted Certificate Authority (CA) certificate
3. The device owner installs the untrusted CA certificate
4. The attacker intercepts the https traffics through a proxy that generates an SSL certificate for each host, signed by its own Certificate Authority (CA) certificate.
5. The attacker can store and analyse the http/https traffic

We reproduce an intercept enrolment https POST:

```
nome=Asked=*******&telefono=*******&
email_masked=****************%40icbp
i.it&email=EMAIL%40EMAIL.IT&password
=PASSWORD&confirmNewPassword=PASSWOR
D&name=NAME&surname=USERNAME&co=&cou
ntry=IT&zipcodeNoModify=20146&provin
ce=MI&city=MILANO&indirizzo=ADDRESS&
telefono_masked=**********&email_mas
ked=*****%40*****&name1=NAME&surname
1=USERNAME&co1=&country1=IT&zipcodeN
```

```
oModify1=20146&province1=MI&city1=MI
LANO&indirizzo1=ADDRESS&telefono_mas
ked=**********&email_masked=*****%40
****&c3=true&c8=true
```

4.2 Access to Data Saved Locally in Wallet

The mobile application that acts as a wallet needs to store information, linked to an account, which enables it to interact with an NFC reader to complete a contactless transaction and to provide the user with summary information via a special user interface.

Storing information in the memory of the operating system, rather than using a Secure Element, has a number of issues, not only if the smartphone is rooted. If local storage is not well protected and access to the wallet restricted, a malicious application could access data such as the payment token provided to the smartphone during Account Provisioning (or later *Replenishment*), and also the credentials used to authenticate with cloud services that serve as the Secure Element, the theft of which would allow the replication of the installation on another device (unless binding has been established between the credentials and the hardware on which they can be used).

A note should be made of users who uninstall the mobile wallet application. In this scenario, all the information previously saved must be safely deleted from the device in order to avoid their discovery and reuse. Figure 4 highlights the part of the architecture vulnerable to the problem.

The issues afflicting Google Wallet (the system for mobile payments developed by Google itself) are an example of this type of attack (Worstall, 2012).

The Android application manager, when uninstalling an application, deletes all data (including the cache) belonging to it (Mulliner, 2009). In the case of Google Wallet, the user's PIN, saved in a common system file, was therefore deleted. The problem lay in the fact that the card data had not been saved in the file system of the phone, but in the Secure Element, and consequently not deleted with all the other data in the wallet.

This allowed anyone to remove the security PIN simply by uninstalling the application, and to set one of their own choice, during reinstallation. Once with a new PIN, the attacker gains access to all the cards previously saved on the Secure Element.

One possible countermeasure is linked to limiting access to critical resources saved locally in the application.

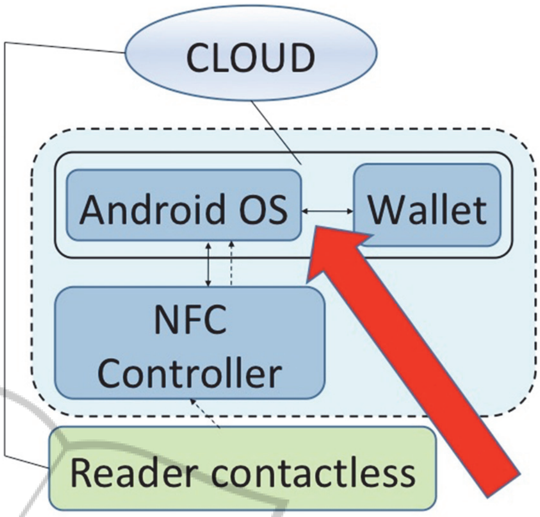


Figure 4: Area subject to vulnerability of data saved in the local wallet.

There have been identified two possible approaches:

a) in the first approach, advantage could be made of storage hardware (supported in Android 4.3 and later) available in some devices. This "hardware-backed storage", also called TrustZone, specifically designed for storing data such as credentials, provides an additional layer of security that makes the extraction of contents impossible since even the kernel of the operating system does not have access.

However, as not all smartphones, at the time of writing, support storage hardware, we have identified the:

b) second approach to mitigating this risk, albeit without completely eliminating it, is to implement mechanisms that supplement the safety features of local storage.

Appropriate, preferably hardware-dependent, encryption would create a link between the data to be protected and the physical device on which it can be used. This approach is recommended especially for the safety of "credentials" (both in the traditional sense and also in forms such as tokens, certificates, etc.) used with the cloud platform. Encryption based on the specific device would greatly mitigate the consequences of the theft of these credentials, because, having created a binding between the credentials and the smartphone, an attacker would also have to subvert the wallet application logic to figure out how to bypass control over compliance with the hardware (Nai-Wai and Li, 2012).

Finally, other mechanisms could be employed to mitigate the risk posed by the use of local storage. In addition to the process of *"tokenization"* (which enables the replacement of the real PAN with virtual, temporary surrogates), the adoption of measures to check the identity of the user (by PINs such as CDCVM or biometric factors), transaction limits (spending limits, number of transactions allowed in one time slot), and runtime checks by the operating system to validate the integrity of the device are appropriate (Hancke et al., 2013).

4.3 Reusability of the Token

The *"tokenization"* process could mitigate some of the risks posed by the HCE solution.

Tokenization can be defined as the process by which the number of a card (PAN) is replaced by a virtual substitute, called "*token*". In fact, a token can be created by emulating the format of the original PAN, so as to be indistinguishable for both a human and also for POS payment terminals, which will treat it as if it were a PAN received in a traditional contactless transaction.

Figure 5 highlights evidence of the part of architecture vulnerable to the problem.

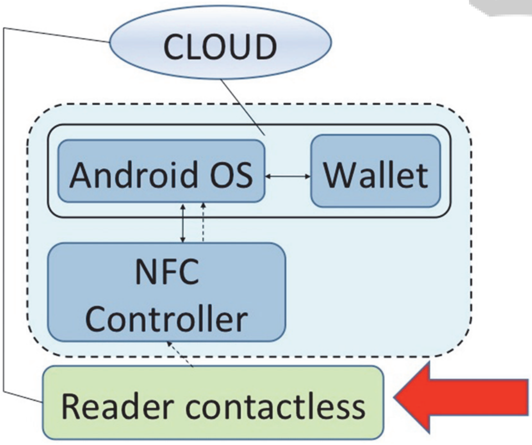


Figure 5: Area subject to vulnerability from "reusability of token".

Tokens must be saved by the application that serves as a wallet on the device, where they are nevertheless at risk. Storing payment tokens rather than the original PAN, however, enables a reduction in the amount of cardholder data in circulation. The security of each token derives from practical impossibility of working out the original PAN.

A further point of attention is given by the choice of using "*single-use*" or "*multi-use*" tokens. Single-use tokens are valid for a single transaction and are thereafter invalidated. They can be used when it is necessary to track the use of a PAN (real or virtual) in multiple transactions, and can be generated, for example, by performing the hash of the PAN with a unique salt linked to the single transaction.

Multi-use tokens do not expire after a single transaction, but can be reused, thus enabling a PAN (real or virtual) to be traced over multiple transactions. This type of token can be generated using the hash of the PAN and a fixed salt for every merchant, but distinct for each user. This being said, an attacker could exploit a number of gaps in the implementation of the system of tokenization to his advantage. Some scenarios are listed below:

- An inadequately encrypted communication channel between the smartphone and the tokenization system would allow both the data of the cardholder and the ties between PAN and virtual tokens to be intercepted by an attacker and used to clone the original card;
- A tokenization system that fails to validate the user's identity, for example through authentication, could be forced to generate new tokens and send them to an untrusted recipient and illegitimate user of the service;
- A process of generating tokens based on weak, no secure cryptographic algorithms are vulnerable to reversing techniques for tracking the original PAN from which the token was created;
- An approach based on "multi-use" tokens that are not specific for single transactions, would enable the spending of the same token multiple times in the event of their interception.

Early evidence of this type of attack occurred in December of 2013 when the company Target was the victim of a widespread attack that led to the theft of data for 110 million users. The stolen data, subsequently made available on the black market, included highly critical information such as names, email addresses, card numbers, expiry dates, CVV codes and PIN numbers. The attack was carried out by malware (called "BlackPOS") that infected POS terminals and exploited the lack of encryption to intercept data received by the POS and forward it to an external server. The attack could have mitigated if the data had been encrypted before reaching the POS terminal.

In a possible attack scenario an attacker could extract the unique id device information (the App stores the user credentials on the storage device) through a malicious App, so:

```
cd /data/data/mobile.app/shared_prefs
cat user.xml
<?xml  version='1.0'  encoding='utf-8'
standalone='yes' ?>
```

```
<map>
<string
name="registrationId">APA91bH7fK9kb40Z
J3CTzWhewtJZd0daOqdYzGKGo86PvSRJXInmNp
bymMffXwn_n1L8y_OntyODQvhilz9oFo54FPCG
QNqQlLBG3yoCg_8Wo_1O4XOYAQXd_KMgd6OB5v
1Fz98LwnqXsrzmAvQGElve-
PLpDihtXA</string>
 <string
name="deviceid">353166057013353</strin
g>
</map>
```

The security and robustness of a system of tokenization mainly depends on the implementation of certain critical components.

Starting from the infrastructure, end-to-end encryption to protect data for its entire life cycle, whilst in transit to the tokenization system and preventing interception of cardholder data or information regarding the PAN-token link, is of critical importance.

The central system should only store the original PAN and encryption keys after prior encryption, so as to be compliant with PCI DSS requirements. New tokens (during card Enrolment or Replenishment) should only be available under certain conditions (such as requiring user verification, or that the sending of the token can only initiated by the issuer).

Therefore, the focus should be placed on two features offered by the tokenization system: that of "token generation" and that of "token mapping". "Token generation" is the process by which a new token is created. The main requirement is that deriving the original PAN from the token should so complex as to be computationally impossible.

This can be achieved by using either a robust encryption algorithm that in turn uses a suitable key or a non-reversible mathematical function (such as a hash function that uses a "salt", or a randomly generated number). In contrast, "token mapping" is the process that maps a token to the original PAN from which it was created.

It is important that, with a view to eliminate (or at least limit) the storage of PANs, the tokenization system should only provide merchants with a virtual payment token and never the real PAN.

Note that if the "multi-use" token is selected, mechanisms to verify and limit use are opportune. These constraints may be based on the following factors:

- Time to live: the period of time in which the token can be spent;
- Number of transactions: the number of transactions that can be performed with the token in question;
- Total amount: the maximum amount of spendable using the token in question;
- Country of use: to allow or inhibit the use of the token.

Note that an attacker cannot modify these transaction constraints, which are generated and verified by the issuer in the central system, even if he is able to install malware on the smartphone of the victim.

The reference standard for payment channels, mail order, telephone and e-commerce is known as the Payment Card Industry Data Security Standard (PCI DSS, 2006-2015). Since companies are constantly at risk of losing sensitive data relating to holders of credit or debit cards, resulting in the possibility of incurring fines, lawsuits and bad publicity, PCI DSS regulatory compliance is one of the main objectives for companies that handle, store, transmit or process credit card data.

PCI DSS standards apply wherever data is stored, processed, or transmitted. Account data consists of cardholder data plus Sensitive Authentication Data, as follows. The PAN is the determining factor in the applicability of PCI DSS standard requirements. PCI DSS requirements are applicable if a PAN is stored, processed or transmitted: if the PAN is not stored, processed, or transmitted, PCI DSS requirements do not need to be considered. If the name of the card holder, the service code and/or expiration date are stored, processed or transmitted with the PAN, or are otherwise included in the cardholder data, such data must be protected in accordance with the requirements of PCI DSS.

The PCI DSS represents a minimum set of control objectives that may be reinforced by laws and regulations at local, regional and sector levels. Moreover, the legislative or regulatory requirements may require specific protection of personally identifiable information or other data elements (for example, the name of the card holder), or define an entity's disclosure practices related to consumer information. Examples include legislation related to consumer data protection, privacy, identity theft, or data security. PCI DSS standards do not supersede local or regional laws, government regulations or other legal requirements.

4.4 Use of a Fake POS

The POS has a key role in the execution of contactless transactions and can therefore be used by attackers to force new transactions or intercept confidential bank data. In fact, when a transaction is initiated, the mobile device generates a contactless transaction that includes various parameters (such as the payment token and its expiry date), which it forwards to the merchant's POS via the NFC interface (Mulliner, 2009). Figure 6 shows evidence of the part of the architecture vulnerable to the problem.

There are a number ways for an attacker to attempt to create a "fake POS" with malicious behaviour:

- **External Fake POS:** using an external device to interface with the smartphone of the victim via communication based on NFC protocol. In fact, if such a fake POS correctly implements the communication flow used by a legitimate POS, the attacker would be equipped to interface with the victim's smartphone, which would be unable to tell the difference between the legitimate and the fake POS.
- **Internal Fake POS:** using a malicious application that interfaces directly with the API of the NFC controller via the operating system. In this case, an exchange of messages via NFC is not even required and the attacker can interact and directly drive the NFC controller by means of the system API.
- **Fake POS Application:** using a malicious application that interfaces directly and attempts to interact with the wallet (if it offers APIs or calling methods of interaction).

Going beyond the specific methods of implementation and focusing the discussion on the possible purposes of attack, first of all a fake POS could be used to stimulate the artificial generation of multiple transactions in rapid succession in order to exhaust the payment tokens stored locally on the smartphone and enable a Replenishment (by which new tokens are supplied to the device). This, in conjunction with an inadequately protected communication channel with cloud (or backend) services would allow the attacker to intercept the information being exchanged (NFC Forum, 2013). Furthermore, if multi-use tokens are used, the attack itself would allow the attacker to accumulate a large number of payment tokens for use later if controls to bind the tokens to the user's identity and/or to the specific transaction are not in place (Hancke et al., 2013).

There are known cases, some of which are very recent in which attackers were able to reverse engineer a contactless payment and identify the instructions and parameters exchanged between a device and an NFC reader. Note that this would permit a reliable reconstruction of the flow of communications for implementation during the creation of a fake POS (Atlassian Bitbucket, 2014).

Of particular note, furthermore, are the security issues raised by a recent study, which stated that it was possible to exploit a flaw in the contact-less payment protocol, which allows transactions in the UK, for example, of up to £20 without requiring the user to enter the PIN (Emms et al., 2014). Moreover, the researchers were able to induce the credit card to transfer $ 999,999.99 in foreign currency to an account owned by the attacker. The attack is based on the use of a malicious POS, hosted on a mobile terminal, which can be pre-set to transfer up to 999,999.99 units in each currency. This terminal can then be exploited for transactions using other smartphones close by, without asking users to enter their PIN.

In order to mitigate the risk posed by a fake POS, it is first of all necessary to prevent the mobile wallet application from exposing interfaces that can be called from other applications and to require the application to check the origin of NFC communications.

The overall risk may also be contained by the use of virtual tokens with limited lifetime (or validity for one single transaction). However, given the absence in the literature of reports of vulnerability and attacks made against cryptogram 10 (CVN 10), together with the fact that cryptogram 43 (CVN 43) has been specifically designed for HCE with CVN 10 as the starting point, the possibility of re-spending tokens appears to be further reduced at the time of writing.

Finally, and on the basis of the mentioned study (Emms et al., 2014) and other research findings (Hancke et al., 2009; Roland et al., 2012; Murdoch and Anderson, 2010), the user's PIN should always be requested in order to avoid the possibility of transactions being carried out without the consent of the cardholder.

4.5 Specific Vulnerability of the Wallet

In the HCE approach, the mobile application acting as a wallet is the heart of the user experience and, being the portion of the infrastructure directly available to the end user, this is where potential attackers will concentrate their efforts.

The first action that can be carried out is the reverse engineering of the application, which, if not well programmed, allows the attacker to identify the presence of any defects that can lead to vulnerability (Haselsteiner and Breitfuß, 2006).

A further critical point is unencrypted log management, which is to be considered non secure as anyone (including other applications) can interface with the smartphone. Therefore, entering confidential information (such as credentials, PIN, or tokens) in the log would frustrate the efforts not only of using a secure communication channel but also those of any secure local mechanisms for saving critical data.

Figure 7a shows evidence of the part of the architecture vulnerable to the problem.

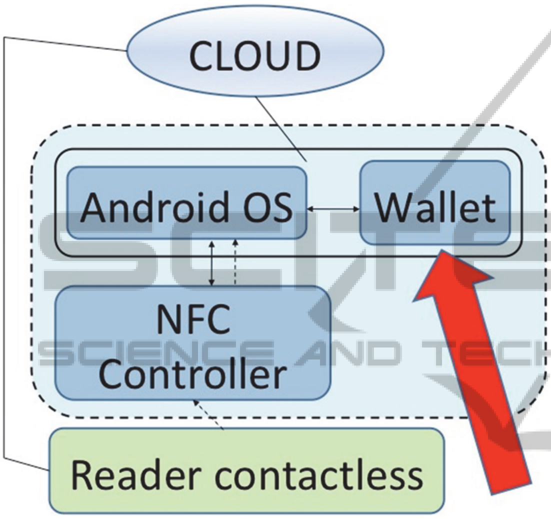


Figure 6: Area subject to specific vulnerabilities of the wallet.

The application must also ensure that the APDU (Application Protocol Data Unit) commands have been received exclusively by the NFC controller; APDU commands received from any other application should be rejected. If this should fail, an application could send malicious commands to the wallet, pretending to be POS terminal, for example.

One attack scenario (called "*Remote Relay*") that takes advantage of the lack of control of the origin of the APDU command was shown at Defcon early as 2012 (Lee, 2012). In this situation, shown in Figure 7a, a malevolent application installed on the smartphone of the victim (*relay*) opens a communication channel and waits for connections from an application mate, i.e. *proxy* (Paya, 2014). When in the presence of a NFC reader, the proxy forwards all APDU commands it receives towards the application relay, which in turn interfaces with the credit card (real/virtual). Therefore, if the wallet does not discriminate the origin of the APDU commands, the victim can be made to make a payment without his knowledge.

In order to mitigate reverse engineering techniques, the application must ensure that the source code, and where possible other assets, are properly obscured.

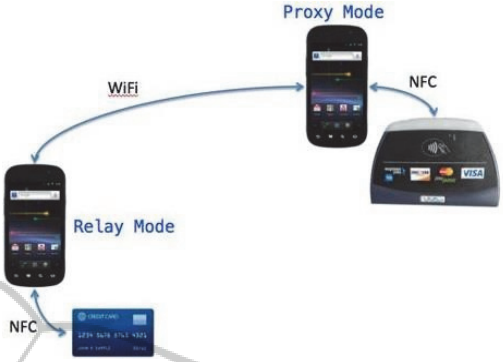


Figure 7a: Diagram of a Remote Relay attack (source: Lee, 2012).

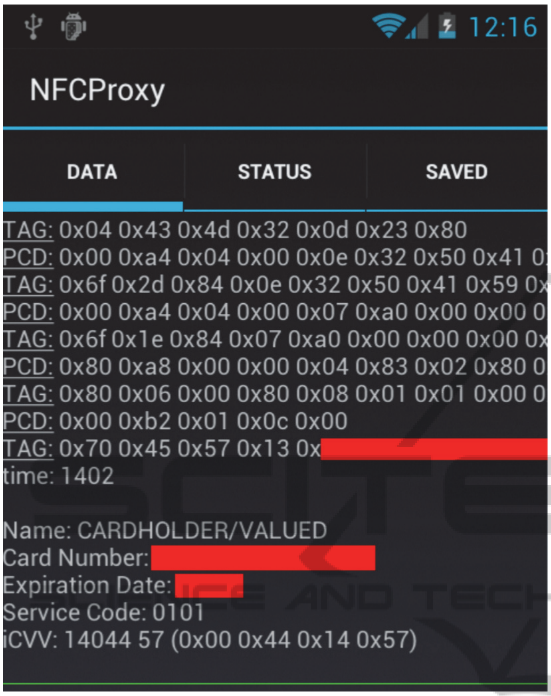
Log management must avoid exposing (directly or indirectly) any credential and/or identity token, and only ever record information that is of no use to a potential attacker, so that whole errors reported by the operating system (such as stack traces) are not saved at all. If necessary, logged information must be encrypted. The application must also include mechanisms to verify its own integrity and that of the device on which it is installed. The application must primarily be resilient to and also report cases of tampering by other malicious applications on the smartphone. Furthermore mechanisms are needed to resist, identify and report if the smartphone is in "*Debug Mode*" or if it has been rooted.

Other techniques to be adopted provide for automatic checks to make sure that the latest version is always available and that the application cannot be installed on removable media (SD card). As previously stated, it is appropriate that the application verifies that APDU commands be received by the NFC controller alone.

Furthermore, software should be developed using best practices and avoiding deprecated methods (or similar) to ensure that the life of the product is compatible with the largest possible number of updates to the operating system. Any functions declared deprecated, should be replaced during the development of updated versions.

Finally, once the application has been developed, practices for code review and analysis of the dynamics of the application are strongly recommended to detect any bugs before release, as is the supply of a secure distribution channel for the application.

In this example, figure 7b, an intercept NFC session payment (Lee, 2012):



```
TAG: 0x04 0x43 0x4d 0x32 0x0d 0x23 0x80
PCD: 0x00 0xa4 0x04 0x00 0x0e 0x32 0x50 0x41 0
TAG: 0x6f 0x2d 0x84 0x0e 0x32 0x50 0x41 0x59 0x
PCD: 0x00 0xa4 0x04 0x00 0x07 0xa0 0x00 0x00 0
TAG: 0x6f 0x1e 0x84 0x07 0xa0 0x00 0x00 0x00 0x
PCD: 0x80 0xa8 0x00 0x00 0x04 0x83 0x02 0x80 0
TAG: 0x80 0x06 0x00 0x80 0x08 0x01 0x01 0x00 0
PCD: 0x00 0xb2 0x01 0x0c 0x00
TAG: 0x70 0x45 0x57 0x13 0x[redacted]
time: 1402

Name: CARDHOLDER/VALUED
Card Number: [redacted]
Expiration Date: [redacted]
Service Code: 0101
iCVV: 14044 57 (0x00 0x44 0x14 0x57)
```

Figure 7b: *"NFC proxy"* (source: Lee, 2012).

4.6 Malware and Fake Application

The presence of malware on the device hosting the mobile wallet application permits various attack scenarios, if the wallet has not been programmed to block local attacks (i.e. attacks from other applications).

As previously stated, a malicious application can attempt to interface directly with the wallet by simulating the NFC controller. This would allow the malware to communicate with the wallet by sending APDU commands, enabling the ability not only to perform *fuzzing* to try to crash the application but also to try to stimulate its functions.

Figure 7b shows evidence of the part of the architecture vulnerable to the problem.

Malware could also have keylogger capabilities to intercept everything that is typed by the user. Especially during the Enrolment phases, if the user were to enter the data to activate the HCE service for a physical card, this would allow an attacker to intercept such data and clone the card, for example. Note that the same risk of access to user data input applies by definition when using alternative terminal keyboards, most of which, however, do not have malicious behaviour. Should an inadequately verified keyboard be used, the risk that the application stores, and forwards confidential information to third parties remains.

Special mention should be made of the increasingly common practice of graphically emulating a banking application to trick users. "Fake apps", as they are known, are able to copy the look and feel of the "original" to induce users to enter confidential data (such as bank details) are becoming increasingly widespread. The Enrolment phase is critical in this case too (Hancke et al., 2013).

It's famous the attack on Santander, which combined fake apps with malware. The malware in question was called "FAKETOKEN", which mimics Santander's token generator. To access the (fake) service, which does nothing but generate an error, the user is prompted to enter the account password, which is immediately sent to a predefined number controlled by the attacker.

4.7 Specific Vulnerability of "Tap & Pay"

"Tap & Pay" is a service offered by Android 4.4 that collects all virtual cards in one simply managed menu on the smartphone. Note that the mobile wallet must also be registered, and a declaration made, as to use of an appropriate AID, and the payment application listed in the "Tap & Pay" menu. The wallet can be set by the user, as the default payment method.

Figure 8 shows evidence of the part of the architecture vulnerable to the problem.

If the "Tap & Pay" system itself has any vulnerability, various issues as described below might be encountered. If the "Tap & Pay" were vulnerable an attacker could control the service by first setting (or resetting) the default payment card against the preferences of the victim.

The attacker could also redirect the target destinations associated with the AIDs As a result he may cause the failure of new transactions (thus causing a "Denial of Service"), if there are incompatibilities in the AID routing rules, or force the operating system to let the user choose which payment application to use, showing a list of possible candidate applications, including malware.

In this case the mitigation of vulnerability in the "Tap & Pay" service can hardly be referred back to the HCE service but rather against Google, as the promoter and developer of the Android platform.

The only precaution is to use the latest updated versions of the Android operating system, which would then include patches to mitigate any vulnerability discovered in previous versions.

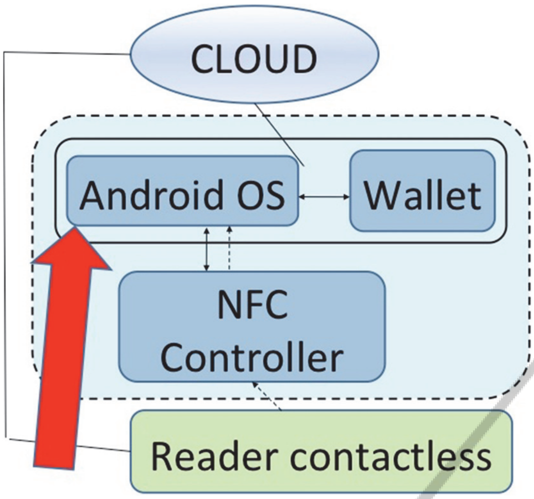


Figure 8: Area subject to vulnerabilities of "Tap & Pay".

4.8 Device Cloud Decoupling

In a standard configuration, the cloud platform interfaces with the smartphone via web services, which are used to enable communication between each.

If the system fails to carry out authentication, or uses static credentials that are detectable by analysing authentication requests (`GET` or `POST`), interfacing with web services directly without using a real wallet would be enabled. This would allow an attacker to use both automated tools (like Nessus) as well scripts created ad-hoc to generate fuzzing and "stimulate" the API to identify any errors in the management of data input, reverse communications, and to reconstruct the flow of communication between the cloud platform and the successfully registered smartphone.

As a result, an attacker could connect to the cloud platform by faking to be a device and attempt operations to which he should not have access.

Mitigation of this type of issue involves securing the route connecting the cloud platform (the "new" Secure Element) to the smartphone by mutual authentication of the parties (machine-to-machine) and encryption of the data in transit.

Note that a critical point is also given by the management of the access credentials to the cloud platform. Card data migration to the cloud means a change in the interpretation of the term *strong authentication*: in this scenario, server access is potentially enabled via simple credentials (username and password), resulting in the loss of the "something you have" factor. In order to re-establish a comparable level of security it is necessary to implement a complementary mechanism that increases the robustness of authentication without compromising user experience, such as the "fingerprinting" of the device allowed to interact with the web services offered by the cloud platform. Moreover, it is appropriate that the login credentials to the Secure Element in the cloud should be entered on a "reasonably" frequent basis, for example at every restart of the smartphone.

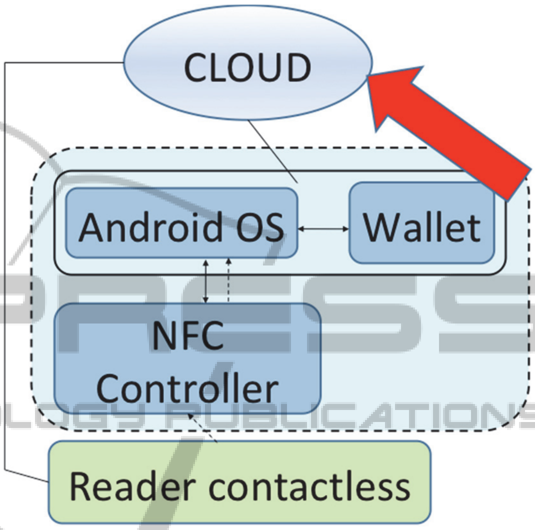


Figure 9: Area subject to vulnerability of "Device - Cloud decoupling".

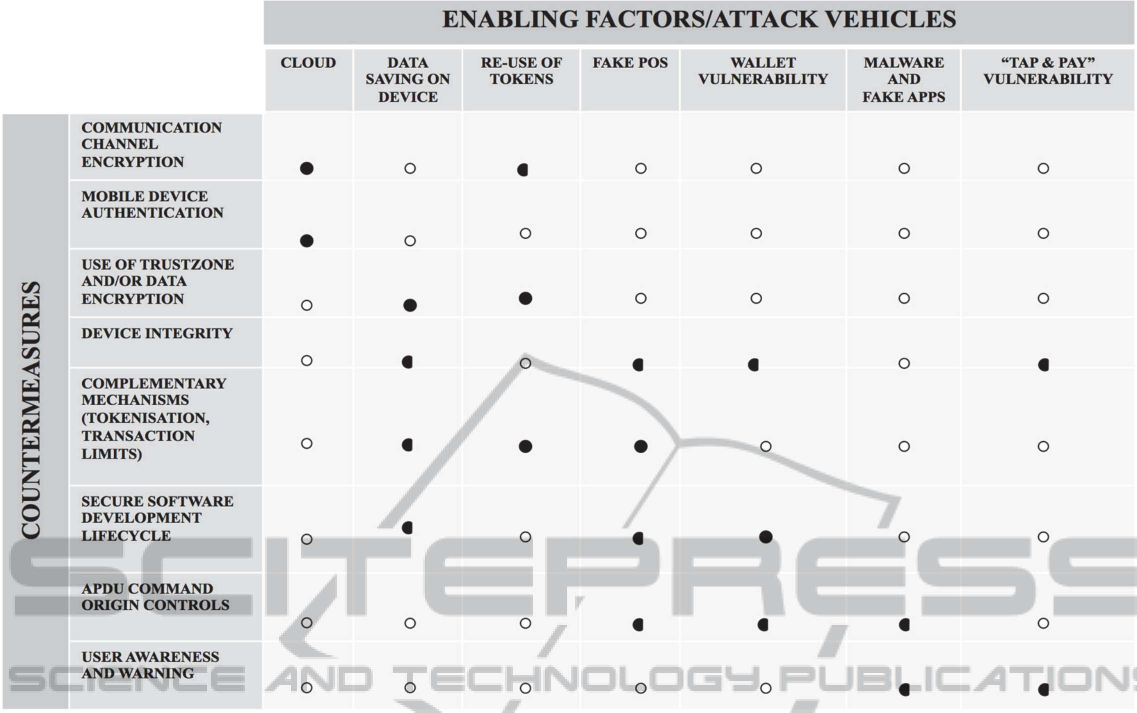
5 CONCLUSIONS

The adoption of the tools and mitigation procedures described in association with the scenarios outlined above would enable an increase in the level of security of a number of aspects of the HCE solution.

The schematic diagram Figure 10 shows the potential level of countermeasure that these instruments, configurations and procedures may enable for each of the areas of attack identified in this chapter

Note also the opportunity to evaluate the use of a "Secure Mobile SDK", a framework to assist the development of the mobile wallet by providing APIs that facilitate the use of security features. These APIs enable the management of, for example:

- root/jailbreak/emulator detection;
- end-to-end encryption;
- machine-to-machine authentication;
- use of secure storage;
- generation of One Time Password (time or location based);
- generation of unique device identifier;



		ENABLING FACTORS/ATTACK VEHICLES						
		CLOUD	DATA SAVING ON DEVICE	RE-USE OF TOKENS	FAKE POS	WALLET VULNERABILITY	MALWARE AND FAKE APPS	"TAP & PAY" VULNERABILITY
COUNTERMEASURES	COMMUNICATION CHANNEL ENCRYPTION	●	○	◖	○	○	○	○
	MOBILE DEVICE AUTHENTICATION	●	○	○	○	○	○	○
	USE OF TRUSTZONE AND/OR DATA ENCRYPTION	○	●	●	○	○	○	○
	DEVICE INTEGRITY	○	◖	○	◖	◖	○	◖
	COMPLEMENTARY MECHANISMS (TOKENISATION, TRANSACTION LIMITS)	○	◖	●	●	○	○	○
	SECURE SOFTWARE DEVELOPMENT LIFECYCLE	○	◖	○	◖	●	○	○
	APDU COMMAND ORIGIN CONTROLS	○	○	○	◖	◖	◖	○
	USER AWARENESS AND WARNING	○	○	○	○	○	●	●

Figure 10: Summary of countermeasures.

- encrypted storage.

REFERENCES

Aigner, M., Dominikus, S., Feldhofer, M., 2007, *"A System of Secure Virtual Coupons Using NFC Technology"*, Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW '07), pp. 362-366.

Atlassian Bitbucket, 2014, *"Reverse engineering of contactless NFC-EMV payments*", <https://bitbucket.org/orbit-burg/nfc-emv/wiki/Home>.

Avison, D., Wood-Harper, T., 2003 *"Bringing social and organisational issues into information systems development: the story of multiview"*. Socio-technical and human cognition elements of information systems. IGI Publishing Hershey, PA (pp. 5-21).

Benner, M.J., Tushman, M.L., 28 2003, *"Exploitation, exploration, and process management: The productivity dilemma revisited"*. Academy of Management Review.

Burgelman, R.A., 47 2002, *"Strategy as vector and the inertia of coevolutionary lock-in"*. Administrative Science Quarterly.

Cavallari, M., 2008, *"Human computer interaction and systems security - an organisational appraisal"*, in: De Marco M., Casalino, N. (eds.), Interdisciplinary Aspects of Information Systems Studies. p. 261-268, Springer, Heidelberg.

Cavallari M., 2011, *"Organisational Constraints on Information Systems Security"*, in: Emerging Themes in Information Systems and Organization Studies, Carugati A., Rossignoli C. (Eds.), 193-207 pp., Springer Physica Verlag Heidelberg.

Devendran, A., Bhuvaneswari, T., Krishnan, A.K., 05 2012, *"Mobile Healthcare System using NFC Technology"*, IJCSI, Vol. 9, Issue 3, No 3.

Emms, M, Arief, B, Freitas, L, Hannon, J, van Moorsel, A, 12 2014, *"Harvesting High Value Foreign Currency Transactions from EMV Contactless Credit Cards without the PIN"*, CCS 2014.

Fiol, C.M., Lyles, M.A., 10 1985 *"Organizational learning"*. Academy of Management Review.

Gupta, A.K., Smith, K.G., Shalley, C.E., 2006, *"The interplay between exploration and exploitation"*. Academy of Management Journal.

Hagen, J.M., Albrechtsen, E. et al., 2008, *"Implementation and effectiveness of organizational information security measures"*. Information Management & Computer Security.

Halgaonkar, P.S., Jain, S., Wadhai, V.M., 2013, *"NFC: a Review of Technology, Tags, Applications and Security"*, IJRCCT, 2013, Vol 2, No 10, 2013.

Hancke, F., 11 2007, *"Radio Frequency Identification, e & i (Elektrotechnik und Informationstechnik)"*, Vol. 124, No. 11, pp 404-408, Springer, November 2007.

Hancke, F., Mayes, K.E., Markantonakis, K., 10 2009*, "An overview of relay attacks in the smart token environment that discusses attack implementations,* *implications and possible countermeasures"*, Computers & Security, Vol. 28, Issue 7, pp 615-627, Elsevier.

Hancke, F., Mayes, K., Mar 2013, *"A Practical Generic Relay Attack on Contactless Transactions by Using NFC Mobile Phones"* In: IJRFIDSC. 2, 1-4.

Haselsteiner, E., Breitfuß, K., 2006,"*Security in Near Field Communication (NFC) – Strengths and weaknesses"*, Proceedings of Workshop on RFID Security (RFIDSec).

He, Z.L., Wong, P.K., 15 2004, *"Exploration vs. Exploitation: An empirical test of the ambidexterity hypothesis"*. Organization Science.

Honig, Z., 05 2013 *"Samsung releases TecTiles 2 NFC tags for Galaxy S 4, available for $15 today"*.

ISO/IEC 14443-3:2011 A&B, <http://www.iso.org/iso/catalogue_detail.htm?csnumber=50942>.

ISO/IEC 7816-4:2013, <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=54550>.

Issovits, W., Hutter, M., 2011, *"Weaknesses of the ISO/IEC 14443 Protocol Regarding Relay Attacks"*, IEEE International Conference on RFID-Technologies and Applications.

JIS-X 6319-4 http://www.proxmark.org/files/Documents/13.56%20MHz%20-%20Felica/JIS.X.6319-4.Sony.Felica.pdf

Juels, A., Syverson, P., Bailey, D., 2005, *"High-power proxies for enhancing RFID privacy and utility"*, G. Danezis and D. Martin, editors, in: *"Privacy Enhancing Technologies (PET)"*, 2005.

Juels, A., Weis, S., 2005a, *"Authenticating pervasive devices with human protocols",* Advances in Cryptology – *CRYPTO*, pages 293–308. Springer-Verlag, Lecture Notes in Computer Science, Volume 3621.

Juels, A., Weis, S, 2005b, *"Defining strong privacy for RFID*", Manuscript.

Lee, E., 2012, DEFCON 20, *"NFC Hacking: The Easy Way"*, ref. NFC proxy, p. 20, http://korben.info/wp-content/uploads/defcon/SpeakerPresentations/Lee/DEFCON-20-Lee-NFC-Hacking.pdf".

Levitt, B., March, J.G., 14 1988, *"Organizational learning"*. Annual Review of Sociology.

Li, Y., Deng, R.H., Bertino, E., 2014, *"RFID Security and Privacy"*, Elisa Bertino and Ravi Sandhu (Eds.), in: Synthesis Lectures on Security, Privacy and Trust, Morgan & Claypool Publishers.

Madlmayr, G., Langer, J., Kantner, C., Scharinger, J., 2008, "*NFC Devices: Security and Privacy*", IEEE The Third International Conference on Availability, Reliability and Security, IEEE DOI 10.1109/ARES.2008.105.

March, J.G., 1991, *"Exploration and exploitation in organizational learning"*. Organization Science.

Marzo, F., Castelfranchi, C., 2013, *"Trust as individual asset in a network: a cognitive analysis"*. In: Spagnoletti, P. (ed.) Organization Change and Information Systems, LNISO vol. 2. Springer, Heidelberg.

Mayes, K.E., Markantonakis, K., Hancke, F., 05 2009, *"Elsevier Information Security Technical Report"*, Vol.14, Issue 2, pp 87-95.

McHugh, S., Yarmey, K., 08 2012, *"Near Field Communication: Introduction and Implications"*, Weinberg Memorial Library, University of Scranton , Scranton, Pennsylvania , USA

Momani, M.H., Hudaib, A.AZ., 2014, *"Comparative Analysis of Open-SSL Vulnerabilities & Heartbleed Exploit Detection"*, IJCSS, Volume (8): Issue (4).

Mulliner, C., 2009, "*Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones*", IEEE International Conference on Availability, Reliability and Security, IEEE DOI 10.1109/ARES.2009.46.

Murdoch, S., Anderson R., 01 2010, *"Verified by Visa and MasterCard SecureCode: or, How Not to Design Authentication"*. Financial Cryptography and Data Security, pp. 42-45.

Nai-Wai, L., Li, Y., 2012, *"Radio Frequency Identification System Security"*, Volume 8, Cryptology and Information Security, IOS Press.

NFC Forum, 12 2013, *"NFC and Contactless Technologies"*, <http://nfc-forum.org/what-is-nfc/about-the- technology>, 2013

Ok, K., Aydin, M.N., Coskun, V., Ozdenizci, B., 2011, *"Exploring Underlying Values of NFC Applications"*, 3rd International Conference on Information and Financial Engineering IPEDR vol.12, IACSIT Press, Singapore.

Ozdenizci, B., Aydin, M. N., Coskun, V., Ok, K., 2010, *"NFC Research Framework: A Literature Review And FutureResearch Directions"*, Proc. 14th IBIMA, Istanbul, Turkey, 2010, pp. 2672-2685.

Patidar, P., Bhardwaj, A., 2011, *"Network Security through SSL in Cloud Computing Environment"*, IJCSIT, Vol. 2 (6) , 2011, pp. 2800-2803.

Paya, C., 05 2014, *"HCE vs embedded secure element: relay attacks (part V)"*, Random Oracle, <https://randomoracle.wordpress.com/2014/05/01/hce-vs-embedded-secure-element-relay-attacks-part-v/>, 2014.

Pettigrew, A.M., 1987, *"Context and action in the transformation of the firm"*. Journal of Management Studies.

Pettigrew, A.M., 2001, Woodman, R.W., Cameron, K.S. *"Studying organizational change and development: Challenges for future research"*. Academy of Management Journal.

PCI DSS, 2006-2015, https://www.pcisecuritystandards.org/security_standards/

Roland, M, Langer, J., Scharinger, J., 10 2012 "*Practical Attack Scenarios on Secure Element-enabled Mobile Devices"*, IEEE 4th International Workshop with Focus on Near Field Communication, 2012. IEEE DOI 10.1109/NFC.

Slade, E., Williams, M., Dwivedi, Y., Piercy, N., 04 2014, *“Exploring consumer adoption of proximity mobile payments”*, JSM, Taylor & Francis, 2014.

Smart Card Alliance, 10 2014, *“Host Card Emulation (HCE) 101”*.

Smith-Strickland, K., 10 2013, *“National Australia Bank Launches Funds Transfer Service Initiated by NFC Peer-to-Peer Mode”*, NFC Times, Oct. 3rd, 2013.

Spagnoletti P., Resca A., 2008, *“The duality of Information Security Management: fighting against predictable and unpredictable threats”.* JISS, Vol. 4 – Issue 3.

Straub, D., Goodman, S., Baskerville, R., 2008, *“Framing of Information Security Policies and Practices”*. In Information Security Policies, Processes, and Practices. D. Straub, S. Goodman and R. Baskerville (eds.), Armonk, NY: M. E. Sharpe.

Suman, S., 09 2013, *“NFC: an overview”,* IJARCSMS, Volume 1, Issue 4, September 2013.

Trend Micro, 01 2015, *“Masque, FakeID, and Other Notable Mobile Threats of 2H 2014”,* http://about-threats.trendmicro.com/us/mobile/monthly-mobile-review/2013-08-mobile-banking-threats>.

Van Damme, G., Wouters, K., Preneel,B., 2009, *“Practical Experiences with NFC Security on mobile Phones”* in Proceedings of the RFIDSec’09 on RFID Security, LNCS, Springer-Verlag, 13 pages, 2009.

Van Dullink, W., Westein, P., 02 2013, *“Remote relay attack on RFID access control systems using NFC enabled devices”*, Report, University of Amsterdam, https://www.os3.nl/_media/2012-2013/courses/rp1/p30_report.pdf.

Verdult, R., Ois Kooman, F., 2011,*“Practical attacks on NFC enabled cell phones”*, IEEE Third International Workshop on Near Field Communication, DOI 10.1109/NFC.2011.16

Worstall, T., 10 2012 , *“Google Wallet's Security Hole*”, Forbes, <http://www.forbes.com/sites/timworstall/2012/02/10/google-wallets-security-hole/>.

Za, S., Marzo, F., De Marco, M, Cavallari, M., 2015, *“Agent Based Simulation of Trust Dynamics in Dependence Networks”*, in: Exploring Services Science, LNBIP, Volume 201, Henriqueta Nóvoa and Monica Drăgoicea (eds.), Springer.