Potential Misuse of NFC Enabled Mobile Phones with Embedded Security Elements as Contactless Attack Platforms

Lishoy Francis, Gerhard Hancke, Keith Mayes and Konstantinos Markantonakis
The Information Security Group, Smart Card Centre,
Royal Holloway University of London,
Egham Hill, TW20 0EX, Surrey, United Kingdom.
Email: {L.Francis, Gerhard.Hancke, Keith.Mayes, K.Markantonakis}@rhul.ac.uk

Abstract

In this paper we investigate the possibility that a Near Field Communication (NFC) enabled mobile phone, with an embedded Secure Element (SE), could be used as a mobile token cloning and skimming platform. We show how an attacker could use a NFC mobile phone as such an attack platform by exploiting the existing security controls of the embedded SE and the available contactless APIs. To illustrate the feasibility of these actions we also show how to practically skim and emulate certain tokens typically used in payment and access control applications with a NFC mobile phone. Although such attacks can also be implemented on other contactless platforms, such as custom-built card emulators and modified readers, the NFC-enabled mobile phone has a legitimate form factor, which would be accepted by merchants and arouse less suspicion in public. Finally, we propose several security countermeasures for NFC phones that could prevent such misuse.

1. Introduction

The recent integration of Near Field Communications (NFC)[1] into mobile devices offers many opportunities for mobile services to incorporate contactless technology. NFC is a short range and standardised wireless communications technology that is being widely adopted to add contactless functionality to mobile devices, e.g. mobile phones and PDAs. This technology allows consumers to perform contactless transactions and connect to peer devices. The NFC device can act both as a card and a reader. This provides the capability for ticketing, banking and other applications, which were historically installed on contactless security tokens, to be implemented on NFC devices. These functionalities allow for NFC-enabled mobile phones to be used as if they were contactless smart tokens, e.g. retail payments at Point of Sale (POS) terminals or swiping an e-ticket at a turnstile, and the opportunity for a single device to contain multiple tokens. The NFC devices can also be used as a contactless reader that interacts with a variety of contactless "smart objects". For example, a customer could initiate the process to buy a cinema ticket by touching his NFC mobile phone against a smart movie poster. More details on the potential of NFC technology can be found in [2].

Up to now, contactless transactions have been well received by merchants and customers due to their inherent benefits such as ease of use, quick transaction time and limited maintenance requirements. However, the increased use of mobile phones in contactless transactions could provide attackers with a new opportunity. It is a possibility that a NFC mobile phone could be used as a contactless attack platform, i.e. provide a suitable hardware device that could allow attacks to be implemented simply by developing suitable software. Contactless token skimmers and emulators currently exist, but a NFC phone platform offers distinct advantages in that it is a small, mobile device and more importantly that is has an acceptable form factor, i.e. it does not physically look like a skimmer or an emulator. In this paper, we investigate the potential misuse of both the token emulation and the contactless reader functionality provided by NFC mobile phones. We first describe the NFC architecture and explain how an attacker could develop software using freely available tools. To illustrate the practical feasibility of misuse we then demonstrate how a NFC mobile phone can be configured as a contactless reader which may be used as a contactless skimming tool, and then how the attacker can use the gathered information to create a "clone" by emulating a token. Based on our experience, we propose countermeasures that would discourage or prevent a NFC mobile phone from being used in such a manner.

2 Background

NFC technology has facilitated the integration of contactless technology into active device platforms such as mo-

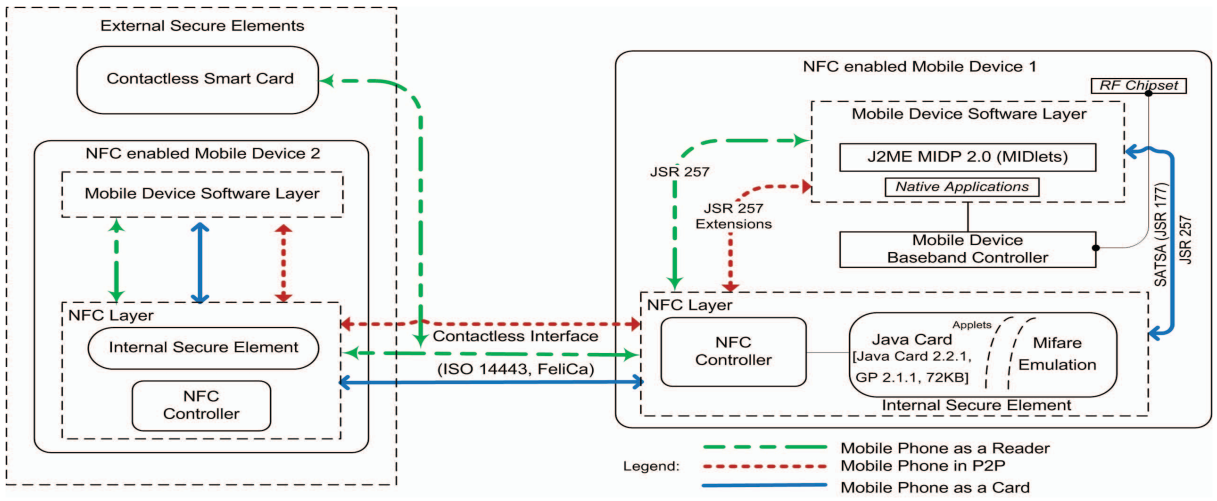


Figure 1. A functional view of the NFC enabled Mobile Phone showing relevant APIs and operational modes.

bile phones. Figure 1 illustrates the functional diagram of the NFC architecture in mobile phones and how it interacts with internal and external components. The core of the NFC system is the Secure Element (SE), which is used to control the NFC based transactions and to establish the trust between service provider and the mobile phone by providing a secure platform for containing sensitive applications and key material. The SE communicates with the NFC controller, the external reader device and the applications installed on the mobile phone through well defined, and standardised, interfaces. There has been a long debate in the industry about the physical form factor of the SE and how it should be interfaced in the mobile phone. Currently there are three architectures that are most often described in literature. The first involves a SE that is an independent embedded hardware module, i.e. an extra smart token IC (Integrated Chip) built into the phone. A second option, preferred by mobile network operators, is to have the existing Subscriber Identity Application module also act as the SE, i.e. to integrate the SE functionality into the Subscriber Identity Module (SIM) [3], Universal Subscriber Identity Module ((U)SIM)[4], Removable User Identity Module ((R)UIM) [5] or Universal Integrated Circuit Card (UICC) [6]. The third option is the use a removable memory component such as Secure MultiMediaCard (Secure MMC) or Secure Digital card (Secure SD) [7] as a SE. The discussion comparing the advantages and disadvantages of these three methods is beyond the scope of this paper, but we should mention that the mobile phones we used, implemented the first NFC architecture with an embedded SE. Regardless of the architecture chosen, NFC allows an enabled device to act as both a "contactless card"

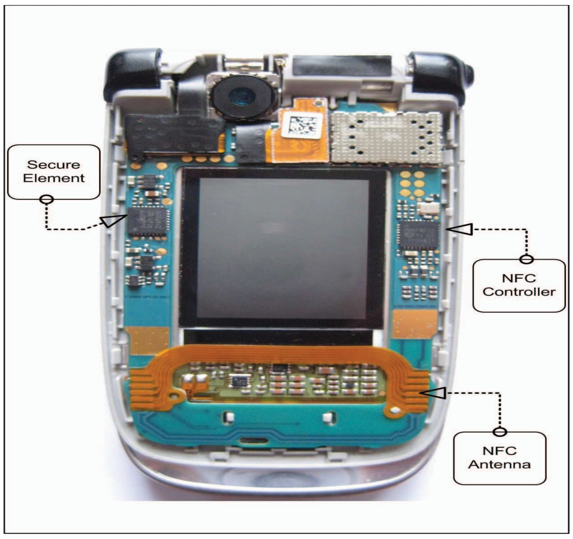


Figure 2. Nokia 6131

and a "contactless reader". In these configurations a NFC enabled phone supports existing contactless communication such as ISO 14443 [8], ISO 15693 [9] and FeliCa [10]. NFC also defines an 'active' communication mode which could be used for peer-to-peer (P2P) communication between two NFC enabled devices. This communication mode, along with the passive token and reader functionality, is further described in ISO 18092 [1] and is beyond the scope of this paper. Now let us look into how a NFC enabled mobile phone could be configured as a "contactless card" and as a "contactless reader" and the role of SE.

2.1 NFC Enabled Mobile Phone as a Contactless Card

The phone, or rather the SE, when configured as a "contactless card" allows for the emulation of passive contactless tokens, e.g. those currently used in payment, ticketing and ID card applications. The SE in the phone is able to act independently as well as interact with applications within the mobile phone's program memory. By default, any NFC enabled mobile phone is a "contactless card" containing pre-installed applications from the device manufacturer. The SE in card emulation mode communicates with the application layer in the mobile phone using the SATSA (JSR 177) API [11] and to the outside external reader over ISO 14443 using ISO 7816 based APDUs (Application Program Data Units) [12] via the NFC controller. The SE could also contain emulation support for other types of tokens using proprietary communication formatting not compatible with ISO 7816 APDUs. If necessary the SE notifies a further application in the mobile phone, such as MIDlet or MIDP [13] application, of any "card" activity triggered by the presence of an external contactless reader.

2.2 NFC Enabled Mobile Phone as a Reader

The NFC enabled mobile phone in contactless reader mode uses JSR 257 API [14] to communicate with any external SE and if needed uses SATSA (JSR 177) API to communicate with the internal SE. The technical literature (e.g. JSR 177) defines an external security element/target as either a contactless smart card or another NFC enabled device. The MIDlet is able to communicate with these security elements by using ISO 7816 based APDUs which are channeled over the established ISO 14443 based connection for the external SE and via SATSA in case of the internal SE. At the application layer, the NFC enabled phone is able to communicate with a smart object such as a smart poster using NDEF (NFC Data Exchange Format) commands via JSR 257 API. JSR 257 also allows communication with non NDEF formatted tags and visual tags. It is worth mentioning that JSR 257 API Extensions allow NFC enabled mobile devices to communicate in P2P mode. The mobile phone we used in our experiments was a Nokia 6131 NFC (as shown in Figure 2)[15]. The SE supports Java Card 2.2.1 [16] (Java Card Open Platform [17]), Global Platform 2.1.1 [18] and Mifare Standard [19] emulation. The SE is implemented on a SmartMX IC and contains 72 kilobytes of EEPROM (Electrically Erasable Programmable Read-Only Memory). By default, the SE on a 6131 NFC is secured or "locked" with the Issuer keyset, which prevents applications from being loaded and installed onto the SE. However, the mobile phone vendor allows SE "unlocking", which sets the authentication keyset to a known public value; further details of the unlocking process can be found in [20]. Applications can now be loaded onto the unlocked SE, but it is no longer seen as trusted by the issuer, i.e. the issuer will no longer install any applications into the SE. At the application layer, the card manager is responsible for managing the applications within the SE and plays the role of the Issuer Security Domain. The reader is advised to refer to [18] for further information on card manager, domains, states, Issuer Security Domain (ISD) and Security Domains (SD). In addition to the support for ISO 14443 based card emulation, the SE supports Mifare Classic emulation. The Mifare Classic is still used in contactless ticketing and access control systems, but uses proprietary communication formats and security algorithms. The JSR 257 API allows MIDlet applications access to the Mifare Standard emulation. The Mifare Standard emulation could also be accessed and managed by the Java Card applets installed within the SE.

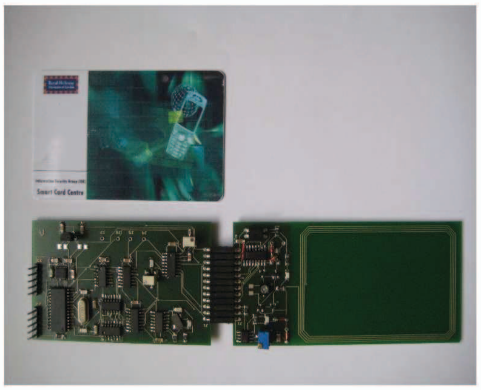
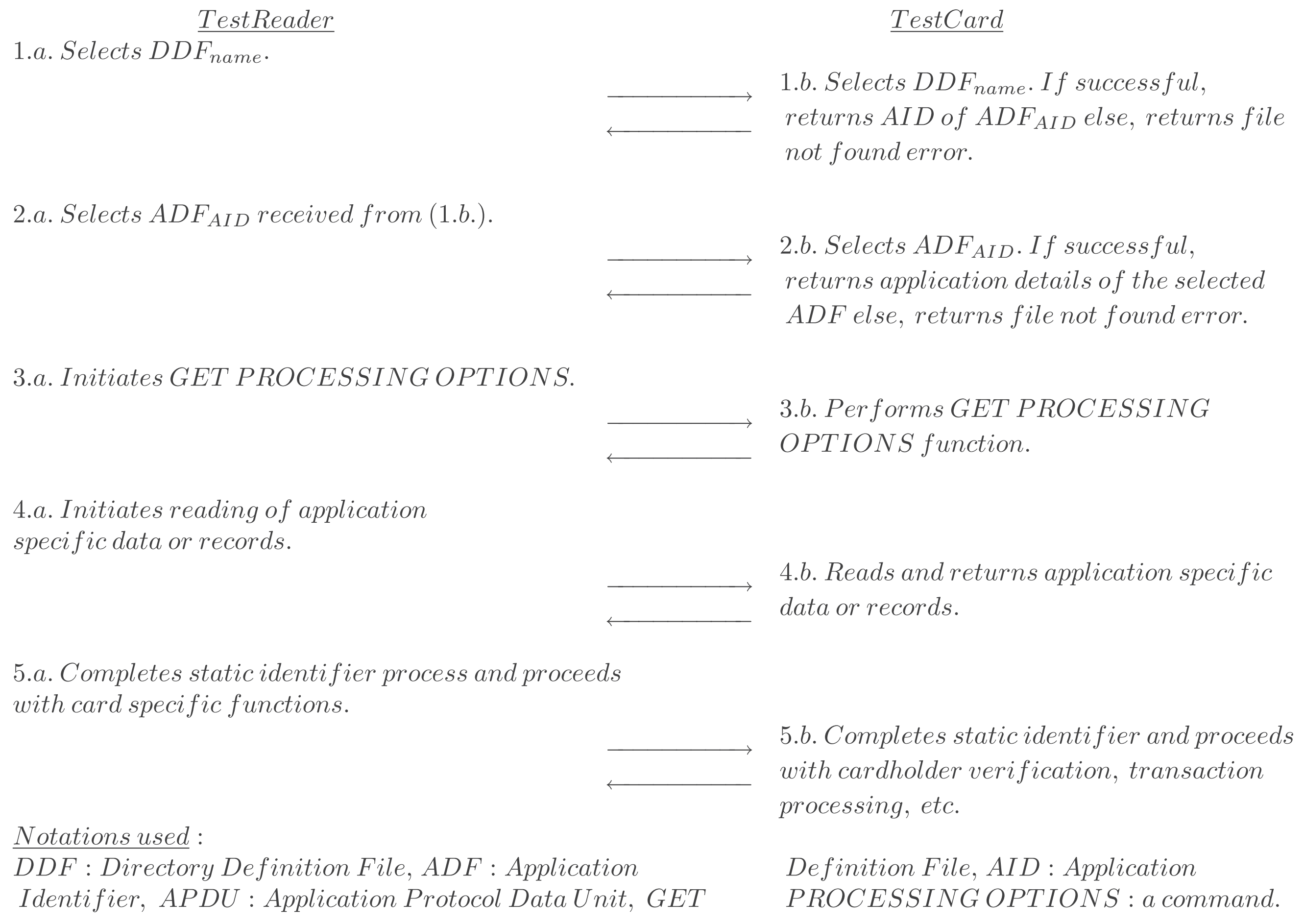


Figure 3. Custom built NFC Card Emulator

2.3 Cloning and skimming attacks

Cloning and skimming are two prominent attacks on contactless systems that are often the subject of practical research. For example, Heydt-Benjamin et.al. describe [21] skimming, replay and relaying on payment systems while there are also examples of unauthorised reading and the subsequent creation of clones, such as e-passport clones [22]. The creation of a clone generally requires a suitable hardware platform and several contactless emulators are published and are available for use. For example, designs for hardware emulators for ISO 14443 tokens can be obtained in [23, 24] and hardware can even be purchased [25, 26]. The major disadvantage of such emulators is that they do not have an acceptable form factor, or in other words they do not look like a legitimate token. An attacker would also need some knowledge and skill in building hardware or be prepared to purchase a assembled device costing up to $750. For skimming attacks the attacker could theoretically use any contactless reader, although these are seldom standalone and portable and need to be connected to a controlling host, such as a laptop, which makes the skimming



TestReader — *TestCard*

1.a. *Selects* DDF_{name}.

1.b. *Selects* DDF_{name}. *If successful, returns AID of* ADF_{AID} *else, returns file not found error.*

2.a. *Selects* ADF_{AID} *received from* (1.b.).

2.b. *Selects* ADF_{AID}. *If successful, returns application details of the selected ADF else, returns file not found error.*

3.a. *Initiates GET PROCESSING OPTIONS.*

3.b. *Performs GET PROCESSING OPTIONS function.*

4.a. *Initiates reading of application specific data or records.*

4.b. *Reads and returns application specific data or records.*

5.a. *Completes static identifier process and proceeds with card specific functions.*

5.b. *Completes static identifier and proceeds with cardholder verification, transaction processing, etc.*

Notations used :
DDF : Directory Definition File, ADF : Application Definition File, AID : Application Identifier, APDU : Application Protocol Data Unit, GET PROCESSING OPTIONS : a command.

Figure 4. Message flow obtained from the Data Capture & Analysis in our test system.

setup cumbersome. One advantage of custom reader hardware is that it has been demonstrated to skim card details at a range greater than the expected operational range of standard contactless systems [27], although this required a suitable power source and larger antennas that could get noticed and compromise the attack. Currently the only device readily available for executing both cloning and skimming attack is the Proxmark3 [28], a portable hardware platform that can both read and emulate contactless cards. This device has become a popular attack/hacking/research tool but with its cost in the region of $450 and illegitimate appearance it is unlikely that it will become a platform for widespread fraud in contactless systems. Our research supplements the mentioned works by looking into the plausibility of mounting the contactless attacks, discussed in the above literature, from NFC enabled mobile phones. Conceptually, the attacks demonstrated in the following sections could work on all NFC architectures such as embedded SE, SE integrated with UICC platform (capable of hosting (U)SIM, RUIM, etc) and Secure Memory, etc. However, we chose to investigate these attacks on a NFC enabled mobile phone with embedded SE.

3 NFC Enabled Mobile Phone as an Attack Platform

In this section, we discuss how an attacker could potentially use a NFC enabled mobile phone to perform two attacks: token "cloning" and contactless skimming. We also describe the implementation of proof-of-concept attacks against a test contactless system that implements only static identification. Our test system is representative of typical contactless systems. For example, access control systems often use tokens responding with only a simple identifier while further examples of real systems can be found that use make use of static cryptographic responses [21]. Our work indicates that both of these attacks can be implemented using a NFC enabled mobile phone. Executing these attacks on a NFC platform poses a significant threat as it is a mobile device with a small form factor, which can easily be used to quickly skim token details. As mobile phones are used more regularly in contactless transactions, a NFC mobile phone attack platform appears as an acceptable token, i.e. it cannot be physically recognised as 'attacker hardware' as it does not look explicitly look like emulator devices (such as those mentioned in the previous section).

3.1 Misusing Card Emulation as a Cloning Platform

In the security attacks discussed in Section 2.3, custom built hardware was needed to emulate a "contactless card". As illustrated in Figure 3, custom built hardware emulators stand out visually and cannot be passed of as a legitimate security token.The cost, time, hardware and technical skill set needed to assemble such an emulator is substantially higher compared with the misused NFC enabled mobile phone, which the attacker may already have or could be bought relatively cheaply, e.g. $200. We were able to emulate a "contactless card" on the mobile device purely at the software application layer and did not have to tamper with any hardware or pre-installed software from the manufacturer. This "cloned" "contactless card" on the phone was accepted as a legitimate contactless token our test system.

3.2 Misusing Contactless Reader as a Pick Pocketing Tool

By developing an application using the standard APIs we configured the NFC enabled phone as a contactless reader that could be used as a mobile "skimming" tool. The custom readers used for skimming attacks, as described in Section 2.3, required significantly high hardware skills, cost and time to build compared to a NFC phone platform. A simple application on the mobile phone MIDlet acted as a contactless reader which retrieved information from legitimate cards that were presented. A disadvantage of using the NFC platform is that the operational range (typically a few centimeters) cannot be easily increased, as was done with some of the custom-built readers. An attacker could, however, compensate for this as the platform is mobile and small. For example, an attacker quickly and covertly reads a token while it is still in victim's possession by waving his/her NFC enabled mobile phone over a contactless payment card.

4 Practical Proof of Concept for the Proposed Attacks

In this section, we demonstrate the proposed security attacks, contactless "cloning" and contactless "pick pocketing". For our experiments, we created a security protocol where an issuer signed static data is exchanged during the authentication process. Our test system consisted of a reader and a "contactless card" that implemented this security protocol.

4.1 Transaction Data Capture & Analysis

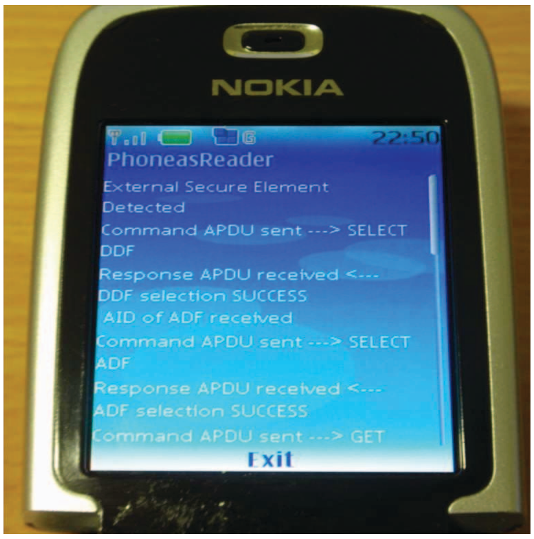
To begin we captured the radio frequency communication between legitimate contactless cards and the reader in our test system. The static messages (as illustrated in Figure 4) required to 'authenticate' the contactless cards were captured by this process.

4.2 Developing the Clone

We "unlocked" the SE which gave us control to load and install applications, i.e. applets. To do this, we downloaded, installed and executed the "unlocking" MIDlet found in [20]. We then developed applets compatible to run on the Java Card 2.2.1 platform of the SE. The Java Card applet was loaded and installed on the SE using a freely available loader, GPShell [29]. These applets behaved like the legitimate token in our test system. We were able to assign the reserved application name to our "cloning" applet and also an additional applet which spoofed ADF (Application Definition File) and having a reserved AID (Application Identifier) of 16 bytes in length. On Java Card platform, the file structure is managed and implemented within the logic of the program codes as Java Card does not support files or file structure. Thus we re-created the application design of the test system environment and other details needed to complete the related protocol. Our applet was designed to receive the standard command messages exchanged in the test system and respond with messages which convinced the reader that it is communicating with a legitimate contactless smart card. Thus, we were able to implement the "clone" (by emulating the behaviour of a legitimate token) on a NFC enabled mobile phone.

4.3 Developing the Mobile Pick Pocketing Tool

We developed a MIDP 2.0 application (which is commonly known as a MIDlet) to emulate a contactless reader using a standard NFC contactless communication API [14]. The MIDlet was developed using a freely available Nokia NFC SDK (Software Development Kit) [30]. The MIDlet was designed to establish an ISO 14443 based connection with external smart cards and exchange APDUs with them. The MIDlet (as illustrated in Figure 5) sends the command APDUs required to extract detailed information of the test system's card which arrived in the form of response APDUs. It was surprising that this API in question, JSR 257, did not require any code signing certificate such as [31], which is usually mandatory for any MIDlet to communicate with smart cards under SATSA API. SATSA API allows a MIDlet to communicate with smart cards such as those hosting SIM, (U)SIM applications present in the mobile phones. In the real world, a hacker without any code signing certificate and with knowledge about the command



```
External Secure Element
Detected
Command APDU sent ---> SELECT
DDF
Response APDU received <---
DDF selection SUCCESS
AID of ADF received
Command APDU sent ---> SELECT
ADF
Response APDU received <---
ADF selection SUCCESS
Command APDU sent ---> GET
```

Figure 5. NFC enabled Mobile Phone as Pick Pocketing Tool

details of a legitimate application could create his own application software to extract details of legitimate contactless cards.

5 Security Countermeasures

In this section, we propose some security countermeasures for preventing the misuse of a NFC enabled mobile phone as a contactless "cloning" and "pick pocketing" platform. As we have seen, we were able to "clone" the contactless application using reserved application names and identifiers. The applets did not require any code signing or verification other than specified by Java Card platform. The SE on which our applets were loaded and installed were easily taken control of by the "unlocking" process as specified by the mobile phone manufacturer. For developing and deploying the "pick pocketing" MIDlet, the communication API did not need any code signing. Even though, the static authentication methods we targeted had inherent weakness that allowed counterfeiting attacks, it is quite concerning that the NFC enabled mobile phones has the potential for becoming a platform to mount cloning and security skimming attacks.

Now let us look into the security countermeasures by briefly considering the timing measurements from our experiments. We measured the timing for each of the command APDUs and the associated response APDUs, which were required to establish a successful transaction. We performed this on two"clones" (one on a NFC mobile and one on a contactless Java Card token) that we created, and on the original card. Table 1, shows the response times of command and response messages for the above mentioned entities. The clones are found to be executing commands with mixed timings. The difference in execution times on the first command was attributed in part to the processing overhead in the application selection (this will be investigated further in future work). The subsequent commands were considered to be more useful for detailed comparison and executed faster in the clones than the original card. We consider that this is due to the increased processing power of the clone operating platform. In general, the execution times of the clones are found to be slower overall, but could be optimised further. So timing based countermeasures may not be feasible against these attacks. We also measured the response times of command and response messages when the phone was used as a "pick pocketing" tool. These measurements indicated that the timing is dependent or dictated by the card application platform and the phone as a contactless reader made no impact on time and behaved normally as any other legacy reader. As no particular security measure is foolproof, we suggest deploying a number of security countermeasures for the proposed attacks and thus filling some of the security gaps we found. Now let us look at them one by one.

- Control measures on the NFC Secure Element (SE):As preventing the "unlocking" of the embedded SE may have many undesired effects such as NFC not being used widely for future applications, we think that placing effective application control measures should be the right approach. The obvious control measure needed is to restrict the usage of the reserved application identifiers on "unlocked" embedded SE or UICC (capable of hosting (U)SIM, RUIM, etc) or Secure MMC/SD based SEs. This could be enforced by the phone manufacturer or the SE issuer or even the mobile network operator. We leave this debate open to the industry. The unlocking process might be made more difficult by the mobile network operator by applying firewalls on their network servers disallowing the completion of the unlocking process. The unlocking process involves establishing a secure HTTP [32] connection with the "unlock" server and downloading the required information. The mobile network operator could apply the firewall rules based on this information. UICC (for e.g. (U)SIM and (R)UIM) based SE on NFC phones would offer greater opportunity to enforce security controls as the network operator itself is the SE issuer; in these tokens are found to have stricter management and security controls in place.

- Making code signing mandatory for NFC communications API: In common with the messaging and security service APIs that are available for use in the MIDlets, the communication API such as JSR 257 could have mandatory code signing controls. This may not

Table 1. Timing Measurements of APDU Command/Responses from Cloning Attack (in milliseconds)

	Original Card	Secure Element	Contactless Smart Card
Command 1	17.444009	236.945944	361.858969
Command 2	14.992275	16.240709	2.194615
Command 3	41.936953	2.893308	1.475977
Command 4	29.595593	3.760029	1.778024
Command 5	16.011098	2.929600	1.494589
Total	119.979928	262.76959	368.802174

stop the misuse, but could discourage the presented attack as the organisation to which this certificate was issued would be made accountable. The strength or weaknesses in the vetting process for applying for code signing certificate is out of scope of this paper.

- Securing the NFC Secure Element activity: The NFC functionality on the mobile phone is designed such that the user is able to control it to a certain extent, such as enabling and disabling tag detection; and controlling the activity settings of the SE. The attacks we presented exploited contactless smart cards and would also work on contactless applications installed on the NFC mobile phones unless the previously mentioned measures are enforced, i.e. the user could set the SE on the NFC mobile phone to "ask first" (requiring a key press) or set it with a pass-code whenever it needs to be active. This would hinder covertly reading by an external reader which could be a skimmer.

- Cryptographically linking the application to Unique Identifiers: The application information could be cryptographically bound to the unique identifier of the legitimate host platform. For example, the unique identifiers could be the ICCID (Integrated Circuit Card Identifier) of the SE or the UID (Unique Identifier) of the NFC controller or the contactless smart card. This prevents an attacker from eavesdropping or reading the data from one token and using it in another platform to create a "clone". Unfortunately, this might not always be possible as there are systems with tokens using random identifiers. For example, some contactless cards return random identifiers during the anti-collision process instead of a UID [33].

6 Conclusion

In this paper, we presented some inherent security issues in the NFC enabled mobile phones with embedded security element. We demonstrated two practical security attacks, using such mobile phones, against a typical contactless application based on static authentication. Although other contactless platforms exist, such as custom-built card emulators and off-the-shelf readers, the NFC-enabled mobile phone has a legitimate form factor, which would be accepted by merchants and arouse less suspicion in public. Furthermore, we proposed countermeasures to inhibit embedded security element based NFC mobile phones from becoming a feasible platform for security attacks such as contactless "cloning" and contactless mobile "pick pocketing". The migration of contactless technology into mobile devices could still improve on the security front where there is a need for more control measures in the host devices that drive the NFC based transactions. If not secured, the Secure Element (SE) embedded in certain NFC mobile phones could become a platform for malicious software. In conclusion, our findings indicate that the embedded SE with the existing security controls and the available contactless APIs could be exploited to configure the mobile phone as a contactless attack platform. These issues needs to be urgently addressed with effective security countermeasures in place.

References

[1] ISO/IEC 18092 (ECMA-340), "Information technology Telecommunications and information exchange between systems Near Field Communication Interface and Protocol (NFCIP-1)". 2004. http://www.iso.org/, Cited 14 July 2009.

[2] "Near Field Communication (NFC) Forum", http://www.nfc-forum.org, Cited 14 July 2009.

[3] "Third Generation Partnership Project, Specification of the Subscriber Identity Module-Mobile Equipment (SIM - ME) interface (Release 1999)", TS 11.11 V8.14.0 (2007-06), http://www.3gpp.org/.

[4] "Third Generation Partnership Project, Characteristics of the Universal Subscriber Identity Module (USIM) application (Release 7)", TS 31.102 V7.10.0 (2007-09), http://www.3gpp.org/.

[5] "Third Generation Partnership Project 2 (3GPP2), Removable User Identity Module (RUIM) for Spread

Spectrum Systems", 3GPP2 C.S0023-C V1.0' (May 26 2006), http://www.3gpp2.org/.

[6] "European Technical Standards Institute (ETSI), Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 7)", TS 102 221 V7.9.0 (2007-07), http://www.etsi.org/, Cited 14 July 2009.

[7] "SD Card Association", http://www.sdcard.org/, Cited 14 July 2009.

[8] ISO/IEC 14443, "Identification cards – Contactless integrated circuit cards – Proximity cards", http://www.iso.org/, Cited 14 July 2009.

[9] ISO/IEC 15693, "Identification cards – Contactless integrated circuit cards – Vicinity cards", http://www.iso.org/, Cited 14 July 2009.

[10] "FeliCa" http://www.sony.net/Products/felica/, Cited 14 July 2009.

[11] JSR 177 Experts Group, "Security and Trust Services API (SATSA) v2.1 for J2ME®", http://jcp.org/aboutJava/communityprocess/final/jsr177/index.html, Cited 14 July 2009.

[12] "International Organization for Standardization, ISO/IEC 7816 parts 1-15", 2005, http://www.iso.org/, Cited 14 July 2009.

[13] "JSR-000118 Mobile Information Device Profile 2.0", http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html.

[14] "JSR-000257 Contactless Communication API 1.0", http://jcp.org/aboutJava/communityprocess/final/jsr257/index.html.

[15] "Nokia 6131 NFC", http://europe.nokia.com/find-products/devices/nokia-6131/technical-specifications, Cited 14 July 2009.

[16] Sun Microsystems, "Java Card Platform Specification v2.2.1", http://java.sun.com/products/javacard/specs.html, Cited 14 July 2009.

[17] NXP, "Java Card Open Platform", http://www.nxp.com/, Cited 14 July 2009.

[18] Global Platform, "Card Specification v2.1.1", http://www.globalplatform.org, Cited 14 July 2009.

[19] NXP Semiconductor. "Mifare Standard Specification", http://www.nxp.com/acrobat_download/other/identification/, Cited 14 July 2009.

[20] "Nokia 6131 NFC Unlocking", http://wiki.forum.nokia.com/index.php/Nokia_6131_NFC_-_FAQs, Cited 14 July 2009.

[21] T.S. Heydt-Benjamin, D.V. Bailey, K. Fu, A. Juels and T. OHare. Vulnerabilities in first-generation RFID-enabled credit cards. Proceedings of Financial Cryptography and Data Security, pp. 1–22, February 2007.

[22] Steve Boggan, 'Fakeproof' e-passport is cloned in minutes. The Times, 6 August 2008, http://www.timesonline.co.uk/tol/news/uk/crime/article4467106.ece, Cited 14 July 2009.

[23] D. Carluccio, T. Kasper and C. Paar. Implementation Details of a Multi Purpose ISO 14443 RFID-Tool. Proceedings of Workshop on RFID Security, pp 181–198, July 2006.

[24] R. Verdult Security analysis of RFID tags. Master Thesis Radboud University Nijmegen, http://www.sos.cs.ru.nl/applications/rfid/2008-verdult-thesis.pdf.

[25] "OpenPCD/OpenPICC Project", http://www.openpcd.org/, Cited 14 July 2009.

[26] HF RFID Demo Tag , TU Graz, http://jce.iaik.tugraz.at/sic/products/rfid_components/hf_rfid_demo_tag, Cited 14 July 2009.

[27] I. Kirschenbaum and A. Wool. How to Build a Low-Cost, Extended-Range RFID Skimmer. Proceedings of 15th USENIX Security Symposium, pp 43–57, August 2006.

[28] Proxmark3: A Test Instrument for HF/LF RFID, http://www.cq.cx/proxmark3.pl, Cited 14 July 2009.

[29] GPShell, http://sourceforge.net/projects/globalplatform/files/, Cited 14 July 2009.

[30] Nokia 6131 NFC SDK, http://www.forum.nokia.com/, Cited 14 July 2009.

[31] "Java Code Signing for J2ME", http://java.sun.com/, Cited 14 July 2009.

[32] "Hyper Text Transfer Protocol", http://www.w3.org/, Cited 14 July 2009.

[33] "MIFARE DESFire EV1 contactless multi-application IC", Product short data sheet, Rev. 02 6 March 2009, http://www.nxp.com/acrobat_download/datasheets/MF3ICD21_41_81_SDS_2.pdf, Cited 14 July 2009.