On the security issues of NFC enabled mobile phones

Lishoy Francis*, Gerhard Hancke,
Keith Mayes and Konstantinos Markantonakis

Information Security Group, Smart Card Centre,
Royal Holloway University of London,
Egham Hill, TW20 0EX, Surrey, UK
E-mail: l.francis@rhul.ac.uk
E-mail: gerhard.hancke@rhul.ac.uk
E-mail: keith.mayes@rhul.ac.uk
E-mail: k.markantonakis@rhul.ac.uk
*Corresponding author

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 an 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. We also discuss how to capture and analyse legitimate transaction information from contactless systems. 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.

Keywords: Near Field Communication; NFC; NFC enabled mobile phones; security threats; skimming attack; cloning attack; secure element; SE; countermeasures.

Reference to this paper should be made as follows: Francis, L., Hancke, G., Mayes, K. and Markantonakis, K. (2010) 'On the security issues of NFC enabled mobile phones', *Int. J. Internet Technology and Secured Transactions*, Vol. 2, Nos. 3/4, pp.336–356.

Biographical notes: Lishoy Francis received his MSc in Information Security at the Information Security Group (ISG), Royal Holloway, University of London, UK in 2004, and BE in Computer Science and Engineering from Visvesvaraya Technological University, Belgaum, India in 2002. Currently, he is a PhD student in the Information Security Group (ISG), Royal Holloway, University of London, UK. His research interests include security of smart card and mobile platforms; security of contactless and proximity technologies; security attacks and countermeasures; security of multi-access system environments; security protocols; security of identification and payment transactions; security evaluation and certification processes; and security prototyping.

Gerhard P. Hancke is currently a Research Assistant with the Information Security Group responsible for the ISG Smart Card Centres RFID and Contactless Research Track and RF/Hardware Laboratory. His main interests are the security of pervasive computing, mobile technology and RFID tokens and the practical implementation of security exploits against secure hardware devices. He has received Bachelor and Masters of Engineering in Computer Engineering from the University of Pretoria, South Africa in 2002 and 2003, and PhD in Computer Science from the University of Cambridge in 2008.

Keith E. Mayes is the Director of the Information Security Group Smart Card Centre at Royal Holloway, University of London. He received his BSc (Hons.) in Electronic Engineering in 1983 from the University of Bath, and PhD in Digital Image Processing in 1987. His current interests are smart card/RFID/NFC security, protocols and applications; mobile communication systems; and transportations systems security and risk assessment.

Kostantinos Markantonakis is currently a Reader in the Information Security Group. He received his BSc (Hons.) in Computer Science from Lancaster University in 1995, MSc in Information Security in 1996, PhD in Smart Card Security in 2000 and MBA in International Management in 2005 from Royal Holloway, University of London. His main research interests include smart card security and applications; secure cryptographic protocol design and analysis, public key infrastructures (PKI), key management, mobile phone security, embedded system security.

1 Introduction

The recent integration of Near Field Communications (NFC) (ISO 18092, 2004) 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 personal digital assistant (PDA). 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 NFC Forum (2010). 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 possible 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

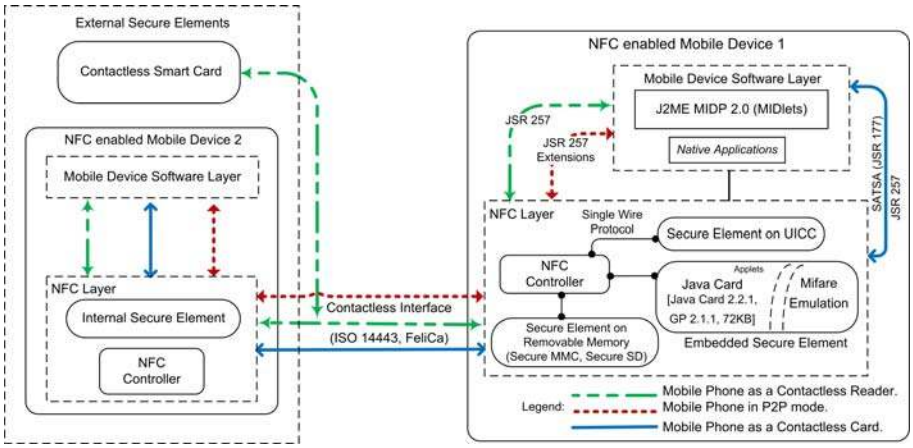
emulators currently exist, but a NFC phone platform offers distinct advantages in that it is a small, mobile device and more importantly that is has an acceptable form factor, i.e., it does not physically look like a skimmer or an emulator. In this paper, we investigate the potential misuse of both the token emulation and the contactless reader functionality provided by NFC mobile phones. We first describe how feasible it is for an attacker to gather transaction information from contactless systems. We also 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 Near Field Communications Framework

NFC technology has facilitated the integration of contactless technology into active device platforms such as mobile 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. Figure 1 shows these architecture options for the SE. The first involves a SE that is an independent embedded hardware module, i.e., an extra smart token integrated chip (IC) built into the phone. A second option, preferred by mobile network operators (MNOs), is to have the existing subscriber identity application (SIA) module also act as the SE, i.e., to integrate the SE functionality into the subscriber identity module (SIM) (3GPP TS 11.11, 2007), universal subscriber identity module ((U)SIM) (3GPP TS 31.102, 2007), removable user identity module (RUIM) (3GPP2 C.S0023-C, 2006) or universal integrated circuit card (UICC) (ETSI TS 102.221, 2007). The third option is the use a removable memory component such as secure multimediacard (secure MMC) or secure digital card (secure SD) (SD Card Association, 2010) 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' and a 'contactless reader'. In these configurations a NFC enabled phone supports existing contactless communication such as ISO 14443 (2008), ISO 15693 (2006) and FeliCa (2010). 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 (2004) 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.

Figure 1 A functional view of the NFC enabled mobile phone showing relevant APIs and operational modes (see online version for colours)



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 (SATSA, 2010) and to the outside external reader over ISO 14443 using ISO 7816 based application program data units (APDUs) (ISO 7816 1-15, 2005) 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 (MIDP 2.0, 2006) application, of any 'card' activity triggered by the presence of an external contactless reader.

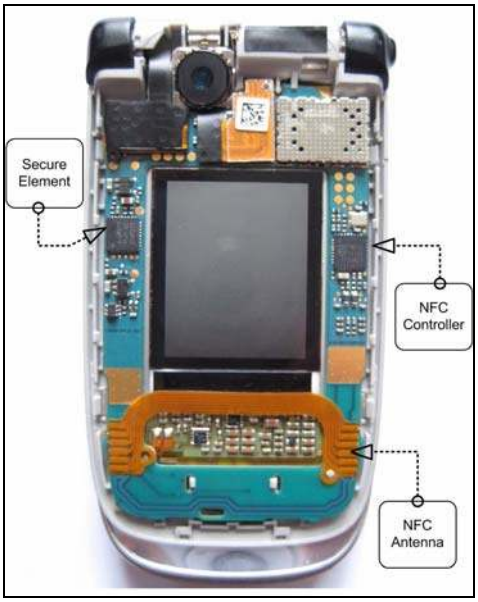
2.2 NFC enabled mobile phone as a reader

The NFC enabled mobile phone in contactless reader mode uses JSR 257 (2006) API 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 channelled 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 NFC data exchange format (NDEF) 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) (http://europe.nokia.com/find-products/devices/nokia-6131-nfc/technical-specifications). The SE supports JavaCard 2.2.1 (2003) [Java card open platform (JCOP, 2010)], Global Platform 2.1.1 (GP 2.1.1, 2003) and Mifare Standard (2010) emulation. The SE is implemented on a SmartMX IC and contains 72 kilobytes of electrically erasable programmable read-only memory (EEPROM). 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 Nokia Forum Wiki (2010). 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 ISD. The reader is advised to refer to (GP 2.1.1, 2003) 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 2 Nokia 6131 (see online version for colours)



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. (2007) describe 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 (Boggan, 2008). 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 Carluccio et al. (2006) and Verdult (2008) and hardware can even be purchased (Open PCD, 2010; TU Graz, 2010). 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 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 (Kirschenbaum and Wool, 2006), 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 (2007), 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. Some vulnerabilities of NFC enabled mobile phones that could be exploited for spoofing of tag content including a proof-of-concept NFC-based ‘worm’ are discussed in Mulliner (2009). 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 Capturing and analysing transactions

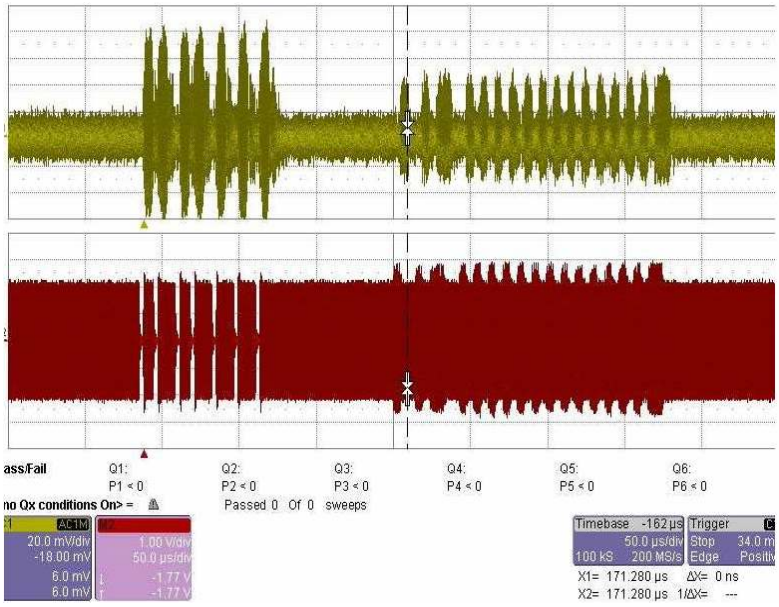
Any attacker who wishes to implement cloning or skimming attacks successfully, first needs to gather information about the target system. The attacker needs to implement an emulator or ‘cloning’ application, whereby his token will talk to a legitimate reader. He therefore needs to know what command the reader will send, and how a real token should respond. The same applies if he wishes to implement a skimming application as he needs to know what commands the token will recognise, and how to interpret the data received. Some systems have public documentation but in the majority of the cases system will be proprietary or not have public documentation. In these cases, the attacker would need to engage in some ‘reverse engineering’ of the system by capturing and analysing the transactions between a legitimate token and reader. The attacker also needs legitimate

data to use in his emulation and skimming applications, such as static authentication message that can be replayed. So after he has implemented the attack platform he still needs to steady the supply of data off legitimate tokens.

This can either be done by once again capturing and analysing transactions or by using the skimming platform to extract information. Due to the inherent nature of contactless technology, eavesdropping is a method that is often discussed as being implemented against contactless systems. As the transactions involved are transmitted over a radio link an attacker is in a position to eavesdrop the transactions, a passive attack that is difficult to detect. Eavesdropping on contactless tokens has been shown to be possible up to a few meters, as the proof-of-concept attacks (Finke and Kelter, 2005; Hancke, 2006) demonstrates. A more detailed examination on eavesdropping is provided in Hancke (2008), which also describes a conceptual design for a inexpensive receiver capable of eavesdropping up to 50 cm as shown in Figure 3(a).

Figure 3 Inexpensive eavesdropping receiver (a) RFSniffer (b) sample trace from RF Sniffer, compared to reader carrier (see online version for colours)

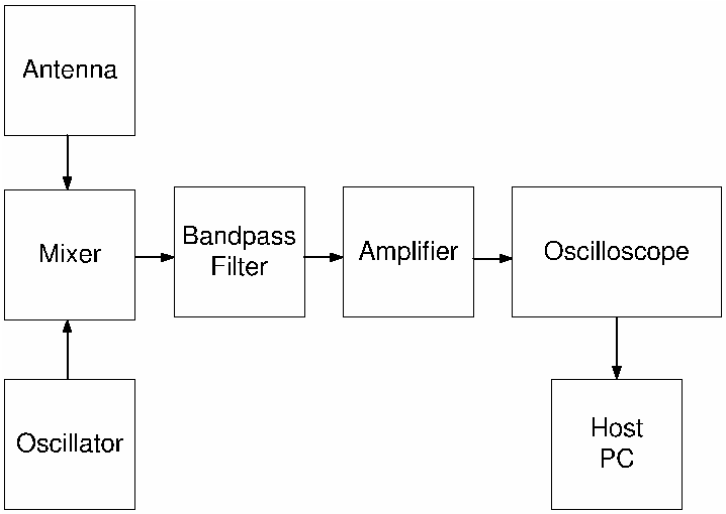
(a)



(b)

One option for capturing the transactions is to use an experimental setup based on (Hancke, 2008), as shown in Figure 4. The attackers simply needs a simple copper coil antenna with resonant circuit to receive the radio frequency carrier, which was then mixed with a secondary sinusoidal signal to move the frequency components to a intermediate frequency of 10.7 Mhz. The mixer, bandpass filter and amplifier are all standard off-the-shelf components and the signal can be captured using an oscilloscope or a analogue-to-digital converter attached embedded processor and some memory. An example of a captured ISO 14443A trace, compared to the actual carrier present at the reader is shown in Figure 3(b). In this example the reader transmits 106 kbit/s Modified Miller encoded data using 3 μs pulses. The forward channel data should therefore be in the first 330 kHz of the spectrum. The token transmits 106 kbit/s Manchester encoded data, which is ASK modulated onto an 847 kHz subcarrier. The backward channel should be in a 424 kHz band centred around 847 kHz. The forward channel is amplitude modulated onto the 13.56 MHz carrier with a modulation index of 100%, while the backward channel has a modulation index of 8%–12%.

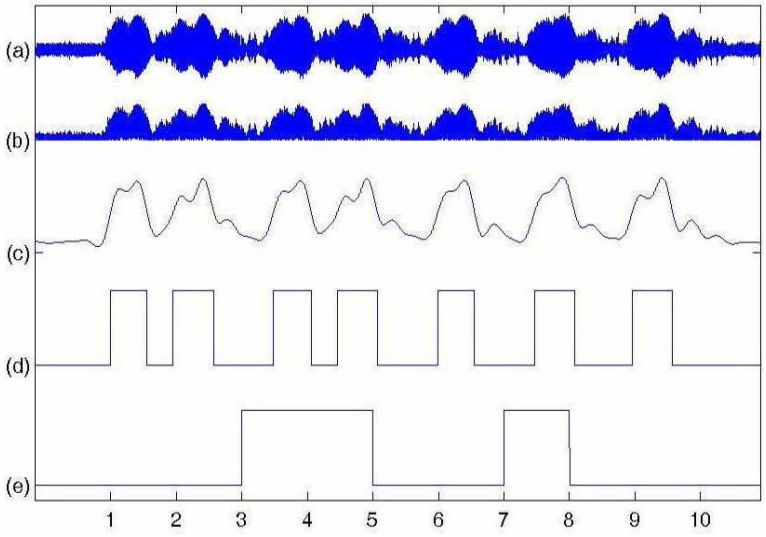
Figure 4 Experimental transactions capture setup



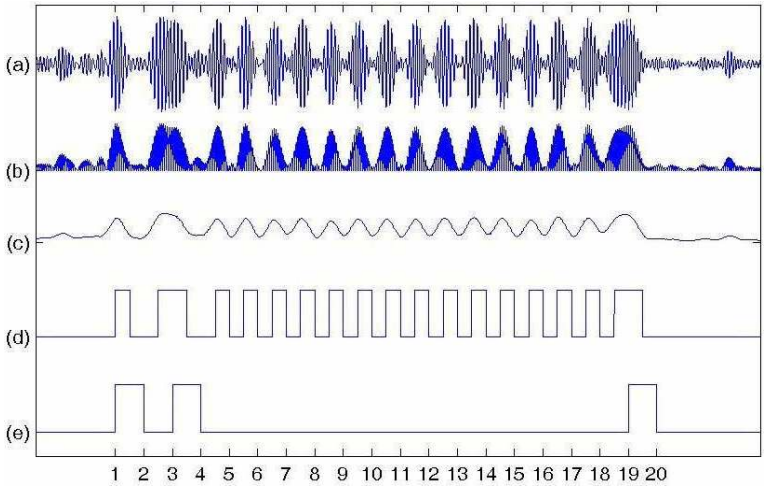
The attacker then needs to analyse the signal to recover the transaction data. This can be done with simple digital processing operations: additional filtering, rectification, correlation and threshold detection. Examples of data recovery for ISO 14443A traces are shown in Figure 5. In Figure 5(a), (a) is filtered again and (b) rectified before being correlated with the base signal (c). Signal (d) is the recovered Modified Miller code and (e) is the final non-return to zero (NRZ) data. In Figure 5(b), (a) is filtered again and (b) rectified before being correlated with the base signal (c) Signal (d) is the recovered Manchester code and (e) is the final NRZ data. This process could be done in embedded hardware or with commercial tools such as MATLAB.

The second option, if the attacker had a larger budget, would be to use a commercial contactless debuggers or testing kits to both capture and analyse the traces. An example of such a tool's recovered data output is shown in Figure 6.

Figure 5 Recovering the transaction data from the captured traces (a) ISO 14333A forward channel (b) ISO 14443A backward channel (see online version for colours)

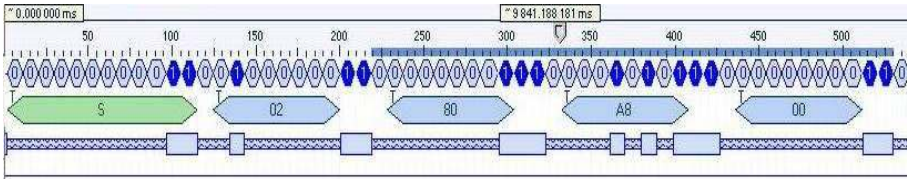


(a)



(b)

Figure 6 Sample trace from a commercial debugging tool (see online version for colours)



4 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 (Heydt-Benjamin et al., 2007). Our work indicates that both of these attacks can be implemented using an 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).

Figure 7 Custom built NFC card emulator (see online version for colours)

4.1 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 7, 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 by our test system.

4.2 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 centimetres) 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.

5 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.

5.1 Transaction data capture and analysis

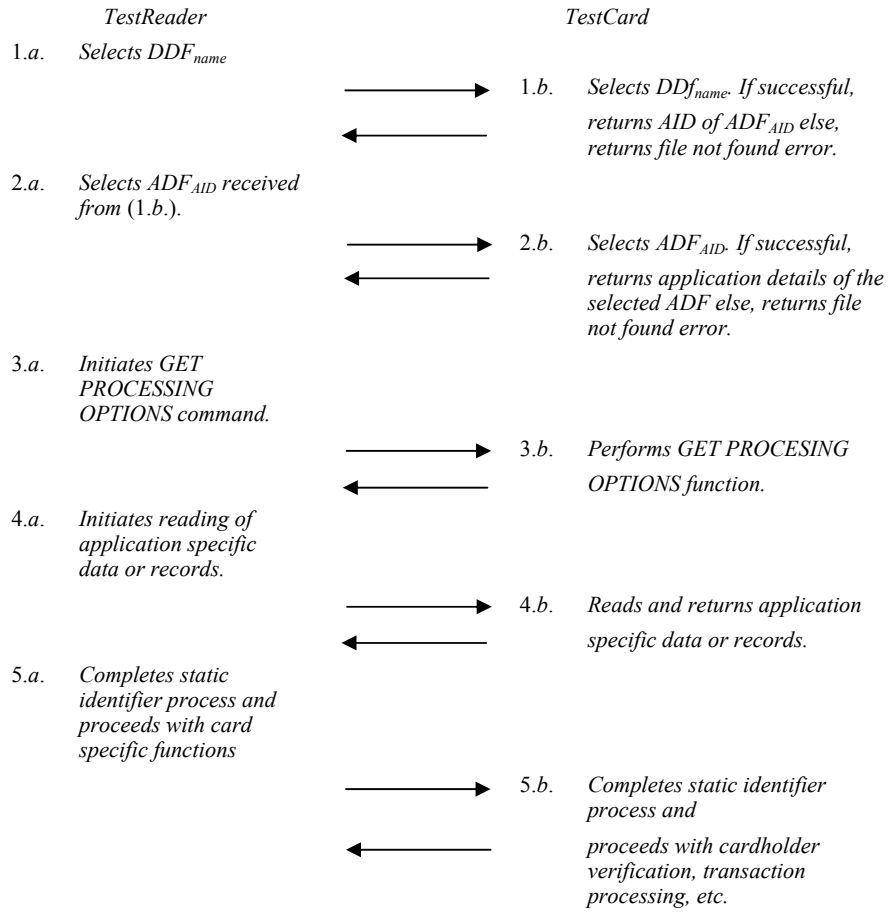
To begin we captured the radio frequency communication between legitimate contactless cards and the reader in our test system. The target/victim (contactless card and the contactless reader) system for the experiment implements a simple static authentication protocol. In such a protocol there is an encrypted dialogue between the card and reader, however because the static data is unchanging, it is vulnerable to record and replay attack. The static messages (as illustrated in Figure 8) required to 'authenticate' the contactless cards were captured by this process.

5.2 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 (Nokia Forum Wiki, 2010). 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 (2008). 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 application definition file (ADF) and having a reserved application identifier (AID) 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.

Figure 8 Message flow obtained from the data capture and analysis in our test system



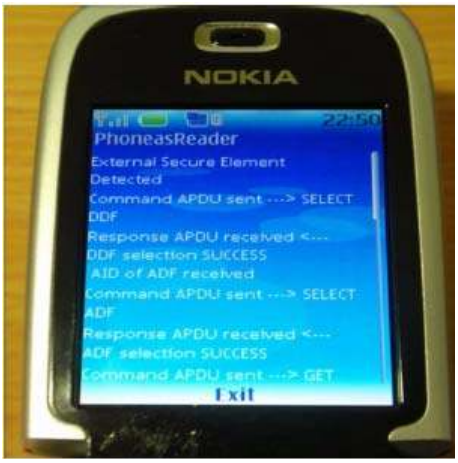
Notes: DDF: directory definition file; ADF: application definition file; AID: application identifier; APDU: application protocol data unit.

5.3 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 (JSR 257, 2006). The MIDlet was developed using a freely available Nokia NFC software

development kit (SDK) (Nokia 6131 NFC SDK, 2010). 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 9) 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 (Java Code Signing, 2010), 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 details of a legitimate application could create his own application software to extract details of legitimate contactless cards. Finally, to complete the experiments both a 'clone' token and 'skimmer' reader were set-up on two separate NFC phones (as illustrated in Figure 10) and they were then able to reproduce the complete original transaction from the legitimate token and the reader.

Figure 9 NFC enabled mobile phone as pick pocketing tool (see online version for colours)



6 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.

Before looking into the security countermeasures in detail, first let us examine the practicalities of security control, ownership and personalisation, of the SE platform as originally carried by Mayes et al. (2010) in the work of the Information Security Group Smart Card Centre. This work is summarised in the subsection 6.1 below and a similar discussion in expanded form can be found in Mayes et al. (2010).

Figure 10 Cloned mobile phone token, skimmer phone and test card (dual-interfaced) (see online version for colours)

6.1 Security control, ownership and personalisation in practice

To address the issues of security control, ownership and personalisation of an SE, we discuss the related security fundamentals. First we need to clarify some assumptions for the hardware functionality that embodies these security fundamentals, as described below.

- Authentication: In order to ensure that entities involved in our trusted solution are legitimate these need to be authenticated. These entities include tokens, reader, phone, server, etc. We refer to contactless smart cards and conventional radio frequency identification (RFID) as tokens.
- Confidentiality: The communication or interaction between authorised entities must be protected from disclosure or discovery by unauthorised entities. This include any information, signals, command messages, command response messages or any functionality of the authorised entities.
- Integrity: Any critical data, software and source code should be protected from unauthorised modification when in storage, operation and deployment.

We assume that the hardware functionality which embodies the above mentioned security fundamentals has been correctly designed, implemented, security-tested and validated. Henceforth, we are not so concerned about the SE chip, but rather its configuration and management. The typical processes in the SE lifecycle management that we will use in our discussion are as follows:

- Initialisation, personalisation and management: With this set of critical processes we mean customising the configuration of the SE prior to issuing it to a customer; and any post-issue changes made to files or functions of SE.
- Migration: By migration we refer to change of mobile network operator (MNO) and/or change of mobile equipment (ME/mobile phone) and/or change of Trusted 3rd party (service provider).

Let us now examine the above mentioned lifecycle processes in detail.

6.1.1 Initialisation, personalisation and management

The conventional Token initialisation is based on the trust relationship between the Token issuer and Token manufacturers. This would usually involve sharing of confidential details such as security algorithms, security keys, and any related sensitive data. In the personalisation process, user specific keys, personal identification number (PIN) and data are generated, loaded onto the token and then validated (Mayes and Markantonakis, 2008). The keys include operational keys and administrative keys that permit any future lifecycle management of the token via the trusted issuer. In the case of Java Card compliant tokens supporting global platform, some post-issue operations such as application loading, deletion, locking and token termination is possible using the required security keys. Since the above mentioned processes are security sensitive, they rely on the integrity and authenticity of the data loaded onto the token. Such critical processes are ideally only carried out in a highly secure environment by an entity adhering to high standards of physical, operational and information technology (IT) security. It appears that the embedded internal SE cannot fit into this standard process and managed by a token manufacturer.

One possible scenario is when the phone manufacturer acts as the issuer and initialises the SE within a secure environment. The challenge in this undertaking is in the personalisation stage, as the phones are normally manufactured and sold without a particular end-user in mind. Nevertheless, the phone manufacturer could follow the MNO approach of handling the SIM, in which case the SE is configured initially. Typically, a SIM is not initially bound or matched to a real customer by MNO. Thereafter, it allows further post-issue personalisation using over the air (OTA) mechanisms. The drawback in this phone manufacturer's approach (and the removable card SE approach) is the difficulty in establishing a trust relationship with the end-user. The end-user is typically not keen on any added inconvenience, complexity or barrier to access the services offered. Unless there is a clear benefit, seamless integration of the SE within the SIM may be preferred. The SIM and/or phone can provide the necessary SE functionality with little added cost, whereas the issuing of additional and personalised memory card SEs could be an expensive option although the later would be able to offer significantly more and cheaper storage.

6.1.2 Migration

There are several scenarios of migration with respect to tokens, mobile phones, networks and services.

- Migration to a new MNO: Currently, when a user migrates to a new MNO the SIM gets replaced. This ensures that the new MNO can offer services based on its algorithms, keys, PINS and other added functions. In replacing the SIM, the user is guaranteed to have hardware and software that have been implemented and tested for security according to the new MNOs standards. With an SE integrated with a SIM solution, migration would require to change the SE (potentially the user account and associated security processes). However, the changes are minimal if the user had relied on a phone with an embedded SE or an SE on memory card option.
- Migration to a new ME: In this scenario, a user relying on SE on SIM and an SE enabled memory card may only be required to change the tokens (inserting the new replacements). However, for any user with an embedded SE on ME the new phone manufacturer would need to have different processes for configuring and managing the associated account.
- Migration to both a new ME and SIM: When the user stays with the same MNO in a scenario where the user changes to a new ME and SIM, there is an opportunity for the solution of an SE in SIM to be seamlessly configured. However, it may need a fresh start with the security and management processes already in place. If the user stays with the same phone manufacturer and having an embedded SE, then just re-configuring the new SE with the settings of the old SE would be sufficient. This would require the SE to be re-personalised outside the original secure environment.
- Migration to a new service provider: Depending upon the appeal of the services, the users may choose to switch to different service providers from time to time. In this case SIM and embedded SE on the phone solutions appear to be better in comparison with memory card SE. This would need re-configuring the memory card-based SE which opens up further security issues related to key management and due to re-personalisation outside of the secure environment.

As evident from the forgoing discussions there is no clear-cut perfect solution for SE provisioning. The SIM card, a conventional and secure token which is personalised in a secure environment, appears to outshine the other options. The cost impact would be significantly lower in order to support the SE lifecycle. With the disparity in implementing the standards and specifications in practice, there are still some risks involved. Although there are sophisticated SIM card specifications (3GPP TS 11.14, 2007) to help with the service provision, almost no mobile phones support them completely. In order to have ubiquitous solutions we need all SIMs to support the NFC SE functionality. The cost conscious MNOs are often tempted to settle for the cheapest SIM cards. This would often allow them to be satisfied with only their minimum requirements (in terms of functionality and data storage) in order to deliver their services.

Perhaps, the major challenge for a 3rd party application developer is that the SIM SE is fully controlled by the MNO. Eventhough the technology exist to open SIM functionality to 3rd parties the track-record of MNOs shows that this did not happen often in practice. This would mean extra effort in working with multiple MNOs for the developers and 3rd party service provider. The solution were the SIM is integrated with SE would need open application management measures, in order to realise its full potential to support NFC-based services. Unless these hurdles are overcome we might have to rely on a patchwork of non-standardised or poorly integrated security measures

and management processes. In moving the solution to the SIM, it would also need to eliminate the 'trust' required between service providers (e.g., banks) and MNOs, as MNOs would have more control of the SE. This need for 'trust' is being minimised by entities called trusted service manager (TSM), who would securely distribute and manage services on behalf of their customers using MNOs, and also act as an intermediary between MNOs and service providers (Hatzmann, 2008; GSMA, 2007). Now that we have discussed the arguments and approaches to SE provisioning, let us now examine the possible security countermeasures for the attacks demonstrated.

6.2 Application security mechanisms

- Using strong cryptographic protocols: The contactless transactions can be secured by strengthening the underlying application protocol by using strong cryptography. For instance, the use of dynamic authentication protocols employing random challenges within the message responses can prevent the attacks presented in this paper.
- Enforcing strict timing: By enforcing strict timing for the transaction messages, the security of the contactless system can be improved. We have observed that the timings vary for the same application when emulated on diverse token platforms. Hence, it is obvious that enforcing total time-based security would be difficult, especially with a variety of security tokens already deployed in the field. 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. 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.
- Radio frequency (RF) shielding: By protecting the security tokens and devices from potential eavesdrop attacks by using RF shielding methods ensures that the tokens would only participate in authorised transactions. However, this would reduce the usability by introducing additional overhead in managing contactless tokens.
- Cryptographically linking the application to unique identifiers (UIDs): The application information could be cryptographically bound to the unique identifier of the legitimate host platform. For example, the unique identifiers could be the integrated circuit card identifier (ICCID) of the SE or the UID 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'. Here we need to assume that the UIDs of the host platform, i.e., the tokens and the devices, are trusted and they cannot be falsely fabricated. By modifying the application protocol to sign and attest the UIDs, would potentially improve the security of transactions involved. One of the potential challenges in implementing such a countermeasure is to incorporate this countermeasure into the application protocol of the existing and deployed systems. Also, this unique binding 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 (NXP Mifare DESFire, 2009).

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

6.3 Security countermeasures on the mobile phone platform

As security is best realised when enforced at all layers of the target platform, let us now examine several security countermeasures on both the mobile phone platform and on the SE platform.

6.3.1 Mobile phone security mechanisms

- 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 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.

6.3.2 SE security mechanisms

- Control measures on the 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 AIDs on 'unlocked' embedded SE or UICC (capable of hosting (U)SIM, RUIM, etc.) or secure MMC/SD-based SEs. This would prevent AID spoofing of applications by the attacker, and this could be enforced by the phone manufacturer or the SE issuer or even the MNO. We leave this debate open to the industry. The

unlocking process might be made more difficult by the MNO by applying firewalls on their network servers disallowing the completion of the unlocking process. The unlocking process involves establishing a secure HTTP (2010) connection with the 'unlock' server and downloading the required information. The MNO could apply the firewall rules based on this information. UICC (e.g., (U)SIM and RUIM) 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.

- Securing the NFC SE 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.

7 Conclusions

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 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.

Acknowledgements

The authors would like to acknowledge the many helpful suggestions of anonymous reviewers and the participants of the 2009 ICITST Conference on earlier versions of this paper. The authors would like to thank Crisp Telecom Limited, UK, for providing equipment support. The authors also thank the editor of this journal.

References

Boggan, S. (2008) ''Fakeproof' e-passport is cloned in minutes', *The Times*, 6 August, available at http://www.timesonline.co.uk/tol/news/uk/crime/article4467106.ece (accessed on 17 February 2010).

Carluccio, D., Kasper, T. and Paar, C. (2006) 'Implementation details of a multi purpose ISO 14443 RFID-tool', *Proceedings of Workshop on RFID Security*, July, pp.181–198.

European Technical Standards Institute (ETSI) (2007) 'Smart cards; UICC-terminal interface; physical and logical characteristics (Release 7)', TS 102 221 V7.9.0 (2007-07), http://www.etsi.org/.

FeliCa (2010) Available at http://www.sony.net/Products/felica/ (accessed on 17 February).

Finke, T. and Kelter, H. (2005) 'Radio frequency identification – abhörmöglichkeiten der kommunikation zwischen Lesegerät und transponder am Beispiel eines ISO 14443-systems', *Bundesamt für Sicherheit in der Informationstechnik*, September.

Global Platform (2003) 'Card specification v2.1.1', available at http://www.globalplatform.org/.

GPShell (2008) Available at http://sourceforge.net/projects/globalplatform/files/ (accessed on 17 February 2010).

GSMA (2007) 'Pay-buy-mobile, business opportunity analysis', Public White paper, November, available at http://www.gsmworld.com/documents/gsma_nfc_tech_guide_vs1.pdf (accessed on 17 February 2010).

Hancke, G.P. (2006) 'Practical attacks on proximity identification systems (short paper)', *Proceedings of IEEE Symposium on Security and Privacy*, May, pp.328–333.

Hancke, G.P. (2008) 'Eavesdropping attacks on high-frequency RFID tokens', *4th Workshop on RFID Security (RFIDSec)*, July, pp.100–113.

Hatzmann, R. (2008) 'Key issues and possible solutions for creating a feasible TSM infrastructure', *Monaco Grimaldi Forum, WIMA-NFC Conference*, April, available at http://www.wima-nfc.com/pics/Image/hatzmann.ppt (accessed on 17 February 2010).

Heydt-Benjamin, T.S., Bailey, D.V., Fu, K., Juels, A., and OHare, T. (2007) 'Vulnerabilities in first-generation RFID-enabled credit cards', *Proceedings of Financial Cryptography and Data Security*, February, pp.1–22.

HF RFID Demo Tag, TU Graz (2010) Available at http://jce.iaik.tugraz.at/sic/products/rfid_components/hfrfid_demo_tag (accessed on 17 February).

Hyper Text Transfer Protocol (HTTP) (2010) Available at http://www.w3.org/ (accessed on 17 February).

International Organization for Standardization, ISO/IEC 7816 parts 1-15 (2005) available at http://www.iso.org/.

ISO/IEC 14443 (2008) 'Identification cards – contactless integrated circuit cards – proximity cards', available at http://www.iso.org/.

ISO/IEC 15693 (2006) 'Identification cards – contactless integrated circuit cards – vicinity cards', available at http://www.iso.org/.

ISO/IEC 18092 (ECMA-340) (2004) 'Information technology telecommunications and information exchange between systems near field communication interface and protocol (NFCIP-1)', available at http://www.iso.org/.

Java Code Signing for J2ME (2010) Available at http://java.sun.com/ (accessed on 17 February).

JSR 177 Experts Group, Security and Trust Services API (SATSA) v2.1 for J2MER® (2010) available at http://jcp.org/aboutJava/communityprocess/final/jsr177/index.html.

JSR-000118 Mobile Information Device Profile 2.0 (2006) Available at http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html.

JSR-000257 (2006) 'Contactless communication API 1.0', available at http://jcp.org/aboutJava/communityprocess/final/jsr257/index.html.

Kirschenbaum, I. and Wool, A. (2006) 'How to build a low-cost, extended-range RFID skimmer', *Proceedings of 15th USENIX Security Symposium*, August, pp.43–57.

Mayes, K. and Markantonakis, K. (Eds.) (2008) *Smart Cards, Tokens, Security and Applications*, Springer Verlag, ISBN: 978-0-387-72197-2.

Mayes, K.E., Markantonakis, K., Francis, L. and Hancke, G. (2010) 'NFC security threats, near field communications, smart card technology international', Courtney, T. (Ed.): January, pp.42–47, ISSN 1361-8288, available at http://www.globalsmart.com/.

MIFARE DESFire (2009) 'EV1 contactless multiapplication IC, product short data sheet', Rev. 02 6 March, available at http://www.nxp.com/acrobat_download/datasheets/MF3ICD21_41_81_SDS_2.pdf)accessed on 17 February 2010).

Mulliner, C. (2009) 'Vulnerability analysis and attacks on NFC-enabled mobile phones', *Proceedings of International Conference on Availability, Reliability and Security, ARES 2009*, March, pp.695–700.

Near Field Communication (NFC) Forum (2010) Available at http://www.nfcforum.org (accessed on 17 February).

Nokia 6131 NFC (2010) Available at http://europe.nokia.com/findproducts/devices/nokia-6131/technical-specifications (accessed on 17 February).

Nokia 6131 NFC SDK (2010) Available at http://www.forum.nokia.com/ (accessed on 17 February).

Nokia 6131 NFC Unlocking (2010) Available at http://wiki.forum.nokia.com/index.php/Nokia_6131_NFC_–_FAQs (accessed on 17 February).

NXP Semiconductor, Mifare Standard Specification (2010) Available at http://www.nxp.com/acrobat download/other/identification/ (accessed on 17 February).

NXP, Java Card Open Platform (2010) Available at http://www.nxp.com/ (accessed on 17 February).

OpenPCD/OpenPICC Project (2010) Available at http://www.openpcd.org/ (accessed on 17 February).

Proxmark3 (2007) 'Proxmark3: a test instrument for HF/LF RFID', available at http://www.cq.cx/proxmark3.pl (accessed on 17 February 2010).

SD Card Association (2010) Available at http://www.sdcard.org/ (accessed on 17 February).

Sun Microsystems, Java Card Platform Specification v2.2.1 (2003) Available at http://java.sun.com/products/javacard/specs.html.

Third Generation Partnership Project (2007) 'Characteristics of the universal subscriber identity module (USIM) application (Release 7)', TS 31.102 V7.10.0 (2007-09) available at http://www.3gpp.org/.

Third Generation Partnership Project (2007) 'Specification of the subscriber identity module-mobile equipment (SIM – ME) interface (Release 1999)', TS 11.11 V8.14.0 (2007-06) available at http://www.3gpp.org/.

Third Generation Partnership Project (2007) 'Specification of the subscriber identity module-mobile equipment (SIM – ME) interface (Release 1999)', TS 11.14 V8.18.0 (2007-06), available at http://www.3gpp.org/.

Third Generation Partnership Project 2 (3GPP2) (2006) 'Removable user identity module (RUIM) for spread spectrum systems', 3GPP2 C.S0023-CV1.0, 26 May, available at http://www.3gpp2.org/.

Verdult, R. (2008) 'Security analysis of RFID tags', Master thesis Radboud University Nijmegen, available at http://www.sos.cs.ru.nl/applications/rfid/2008-verdult-thesis.pdf.