NFC Security Threats

Keith E. Mayes, Konstantinos Markantonakis, Lishoy Francis, Gerhard Hancke
Information Security Group Smart Card Centre
Royal Holloway, University of London
Egham, Surrey, TW20 0EX England
(keith.mayes, k.markantonakis, l.francis, gerhard.hancke) @rhul.ac.uk

Abstract

Near Field Communication is undoubtedly an interesting technology that can open the way to new applications for the benefit of users and service providers. Most anticipated services involve some aspect of security and privacy protection and so it is important that NFC systems provide adequate protection to sensitive data, functionality and communications. In this paper we discuss the security challenges posed by the current Security Element (SE) hardware architectures within the mobile device, SE ownership and management issues and reliance on the radio interface. Furthermore we discuss how NFC may become a favoured tool for attackers and illustrate this by means of a practical example.

1 Introduction

Mobile communication and computing technology has made astonishing advances since its inception in the 1980s. In fact today's mobile phone has more in common with an advanced computing platform than the humble telephone that preceded it. In its meteoric rise to ubiquity, the mobile phone also helped to establish the smart card in the form of the Subscriber Identity Module (SIM) [1]. Contactless smart cards, which are basically smart card chips combined with Radio Frequency Identification (RFID) [2], have also been making advances and exist in diverse forms and with various functional and security capabilities. For convenience we will refer to contactless smart cards and conventional RFIDs collectively as Tokens. In many cases, the desire for ease-of-use and reliability has led to a prevalence of contactless interfaces between Tokens and Reader devices. As a result, the form factor of the Token and the Reader (to a lesser degree) becomes flexible to the extent where various objects could be made to act as Tokens. This is indeed the case with Near Field Communication (NFC) [3] interfaces in mobile phones. Depending on your viewpoint, NFC can be exciting because the phone can act as a Token, as a Reader, or just enable peer-to-peer communications between mobiles. Excitement is normally tempered by practicality and in the case of NFC by the need to ensure adequate security protection. If we need to read Tokens purely for intuitive and fun user applications, such as retrieving a website address from an object, then security may not matter as much, but the Tokens could include bank cards, passports, e-IDs, health/entitlement cards and travel tickets. As a result, we become concerned about information and transactions that communicate sensitive information and are of "real-value". Conventional Readers are normally specially designed to resist a range of physical, logical and side channel attacks and then to operate in a managed (or at least supervised environment) which is not the case for mobile phones. It is therefore important to build the mobile platform into as much of a secure Reader as possible, as otherwise its vulnerability to attack will simply undermine any application. Similarly, Tokens (except the simplest of RFIDs) are normally designed to resist attacks in an effort to prevent discovery or modification of sensitive information and to decrease the potential for the creation of unauthorised clone devices. An NFC phone must therefore have equivalent attributes when used in place of a conventional Token. In this paper we will investigate how we might build this security, although at this point we will note that an NFC enabled phone that can act as a Reader or Token emulator has its uses even when lacking attack resistance. Unfortunately some of these uses are of benefit to security attackers rather than legitimate operation. In fact there is a great deal of interest in attacking Token systems and indeed it has become somewhat of a "sport". It typically involves building equipment to eavesdrop on transactions and/or emulate legitimate Tokens or Readers. However the NFC phone has the potential to offer some functionality of this attack equipment in a cheap, sophisticated and off-the-shelf package. Indeed if it becomes commonplace for users to present their phones to Readers instead of conventional Tokens it could be very hard to spot an attacker. In principle it may be possible to control phone security so that the platform cannot be misused by attackers, but as we will describe, the management of security is not yet well handled and there are ownership, control and personalisation issues that threaten an optimum solution. Finally we will describe a practical and published attack that demonstrates how the current fragmented security control could be exploited in a cloning attack. We will begin by considering the NFC security architecture options within a typical mobile phone.

2 Security Elements within the NFC Architecture

The need for security in NFC transactions is not a new idea and it is meant to be provided by a security controller [4] in the form of a Security Element (SE). The SE is intended as an attack resistant microcontroller, rather like the chip found in a good quality smart card. However because the SE is pivotal in NFC transactions and ownership/control of it may yield commercial or strategic advantage, various solutions have been promoted. A general idea of the options can be appreciated from Figure 1.

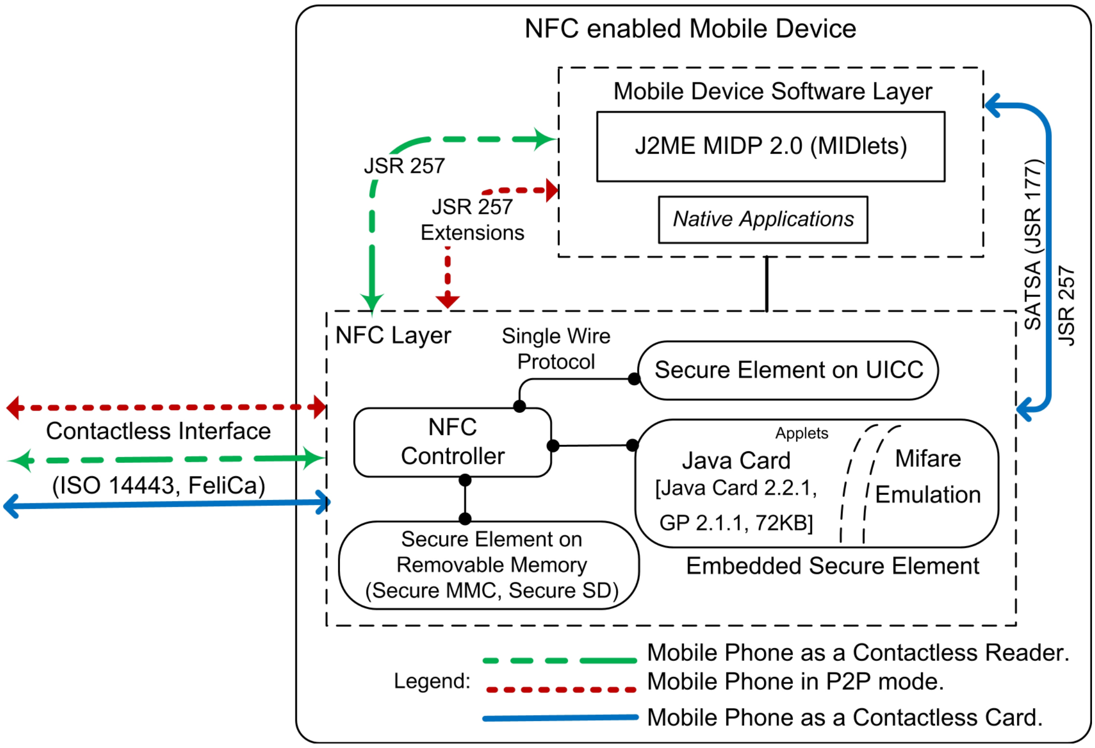

Figure 1 NFC Security Architecture Options

Figure 1 shows at least 3 options. Firstly (and common in legacy NFC phones) the SE is a separate internal chip built onto the Printed Circuit Board (PCB) of the phone. Who really owns it is a good question, but it is likely to be controlled by the mobile phone manufacturer. Another possibility is to put the SE on a removable memory card in which case it could be owned by a third party. Last but not least, is to remember that the phone already has a hardware security module in the form of the Subscriber Identity Module (SIM) implemented on a UICC and this could also include the additional functionality needed for the SE, albeit under the control of the issuing Mobile Network Operator (MNO). From a technology point of view all the options are conceptually similar i.e. a secure microcontroller connected in a variety of ways to the PCB, however as we shall see there can be some major differences when we start to think about personalisation and secure management of the SEs.

3 Practicalities of Security Control/Ownership and Personalisation

Before addressing the issues surrounding secure control/ownership and personalisation of an SE it is worth summarising the kind of functional security fundamentals that we are looking for.

- Authentication
  - To ensure that entities (Token, Reader, Phone, Server etc) involved in our trusted solution are legitimate/authentic.
- Confidentiality
  - Information, signals, commands or functionality that are restricted to certain authorised entities must be protected from disclosure/discovery by unauthorised entities.
- Integrity
  - Critical data and applications code should be protected from modification when in storage, operation or during communications/transactions.

For the purposes of this paper we will assume that the SE satisfies the following conditions.

> *The hardware functionality that embodies our security fundamentals has been correctly designed, implemented and tested to strongly resist the anticipated attacks that may be made against it.*

In which case we are not so concerned about the SE chip, but rather how it is configured and managed. Below are some typical management processes in the SE's lifecycle that we will use in our further discussions.

- Initialisation, Personalisation and Management
  - Customised configuration of the SE prior to issue to a customer
  - Changes to files and functions after issue to a customer
- Migration
  - Change of MNO
  - Change of Mobile Equipment (ME/phone)
  - Change of Trusted 3rd Party

3.1 Initialisation, Personalisation and Management

For conventional Tokens, Initialisation is strongly protected under the trust relationship between the Issuer and Token manufacturers, which may involve sharing of confidential details of algorithms, common applications, common keys and sensitive data. Personalisation is the next critical step (and often trusted to the Token manufacturer), as user specific keys, Personal Identification Number (PIN) codes and data are generated and loaded onto the Token [5]. Aside from operational keys, there are usually administrative keys and PINs to permit future lifecycle management of the Token via the Issuer. In the case of Java Card [6] compliant Tokens supporting Global Platform [7], the key(s) may also permit operations such as application loading, deletion, locking and Token termination. Clearly, initialisation, personalisation and the associated key generation and management processes are extremely security sensitive operations and reliant on the integrity of the information loaded onto the Token. These critical processes establish the foundation for the Token's administrative and operational security, and are ideally only carried out in a secure environment by a party adhering to the highest physical, operational and IT security standards. However, considering our options for the SE in a NFC phone, it appears that the embedded/internal SE cannot fit this standard process. If we consider that the phone manufacturer (perhaps via a trusted third party) also acts as the Issuer then it could initialise the SE within a secure environment, however personalisation is still a problem as phones are normally manufactured and sold without a particular end-user in mind. The manufacturer could however configure each SE with a unique account that is not matched to a real customer, which is similar to the approach of MNOs when issuing SIM cards for network access and then allowing further personalisation at a later stage using Over The Air (OTA) mechanisms. This approach (and the removable card SE) would necessitate another trust relationship for the end-user who is typically not keen on any added inconvenience, complexity or barrier to services, and so unless there is a clear benefit there may be a preference for a seamless integration of the SE within the SIM. There are also cost considerations and the issuing of additional and personalised memory card SEs could be an expensive option, especially when the SIM and/or phone can provide the necessary SE functionality at little added cost.

3.2 Migration

Migration is an important consideration for mobile phones, networks and services and there are a number of scenarios that deserve consideration.

- Migration to a new MNO is usually achieved simply by SIM replacement. This ensures that the stored data, algorithms, keys, PINS and added functions are exactly as required for the new MNO. Replacing the complete SIM card also provides assurance that security critical functionality and storage plus the associated attack resistant countermeasures in the underlying hardware and software have been implemented and tested according to the MNO's standards. For an integrated solution this type of migration means that we have also changed the SE and potentially the user account and security management processes. If however the user had relied on the phone or memory card SE then little may have changed.

- Migration to a new ME may be achieved by inserting the original SIM and optionally the SE enabled memory card. If the user was reliant on the SE embedded in the phone it will have changed completely and the new manufacturer may have different processes for configuration and management of user accounts. By contrast the SIM and memory card SE options could be largely unaffected.

- Increasingly, migration can mean both a new ME and SIM. If the user stays with the same MNO there is a good chance that the SE within the new SIM will be seamlessly configured, otherwise it will be a case of starting again. Similarly, staying with the same phone manufacturer might result in the new embedded SE being reconfigured with the settings of the old one, although this is likely to require re-personalisation of the SE outside of a secure environment.

- Migration is often triggered when a user finds a more advantageous service provider and so this could also happen with the memory card SE, in which case the SIM and mobile phone based solutions appear better options. In principle a memory card SE could be reprogrammed, but this raises significant security questions about key management and re-personalisation outside of secure environments.

From the forgoing discussions one can see that in the case of SE provision there is no clear-cut perfect solution. The most promising option appears to be the SIM card, as it is a conventional and secure Token that is normally personalised in a secure environment and the SE support should have little impact on cost; which keeps the SIM as a “throwaway” device. In fact, this is the way that standards and opinions are moving, although there are still risks involved. Just because something is standardised does not mean that everyone will implement and/or use it. For example, there are sophisticated SIM card standards [8] that would be very helpful in service provision, but almost no mobile phones support them completely. Furthermore if we want ubiquitous solutions we need all SIMs to support the NFC SE functionality, whereas cost conscious MNOs are often tempted to buy the cheapest SIM cards that satisfy their minimum requirements for functionality and data storage. Perhaps the biggest problem for an application developer or third party service provider is that the SIM/SE would be completely controlled by the MNO. Not only does this mean the extra effort of working with multiple MNOs, but MNOs have never been very good at opening their SIM card functionality to third parties, even though the technology exists to do this. Therefore, although the SIM with embedded SE has potential to fulfil NFC security needs, it may well be by-passed for application management reasons; returning us to a patchwork of non-standardised or poorly integrated security measures and management processes. Not only does this mean that our NFC solutions will not be as secure and/or useable as we would like, but the NFC phone platforms could become a favoured tool for hackers, as we will describe in the following example.

4 Misusing NFC Phones

We have discussed the differing arguments and approaches to SE location, ownership and management and we will now show how a legacy phone embedded SE could be exploited in a skimming and cloning attack. This work [9] was originally carried out in the ISG Smart Card Centre (SCC) at Royal Holloway University of London. The test phone is shown in Figure 2 with its cover removed in order to show the physical components referred to earlier. Some skimming and cloning attacks e.g. [10], may seek to build special hardware or physically modify existing hardware components, but the attack we describe requires only software development.

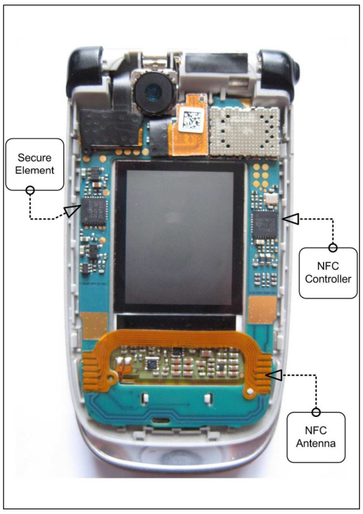

Figure 2 Exposed view of NFC phone

The target/victim system for the experiment is a Token (contactless smart card) and Reader implementing a simple static Token-data authentication protocol. In such a protocol there is an encrypted dialogue between the Token and Reader, however because the static Token-data is unchanging, it is vulnerable to record and replay attack. The first step is to capture the message flows from a valid transaction. With contact card systems this would have been rather difficult, but contactless systems use radio communication that can be eavesdropped. The equipment needed for this is not expensive and a basic "sniffer" can be easily constructed [11] and connected to a digital oscilloscope. Figure 3 shows a low-cost sniffer and a typical capture/trace.

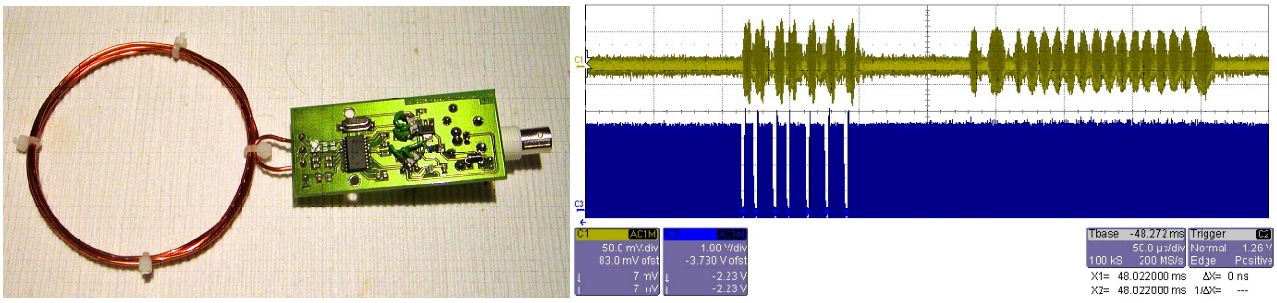

Figure 3 RF Sniffer and Oscilloscope Trace

After processing on a standard PC we have an exact record of a legitimate transaction. The first attack to be considered uses the NFC phone to act as a clone Token platform. We first "unlocked" the Java based SE, via a downloaded MIDlet (phone application), which gave us control to load and install applications on the SE. We then developed, loaded and installed applets on the SE. We were able to assign the reserved application name to our "cloning" applet and also created an additional applet with a reserved Application Identifier (AID), which spoofed the Application Definition File (ADF). Our applet was designed to receive the standard command messages exchanged in the target system and respond with messages which convinced the Reader that it was communicating with a legitimate Token. Thus, we were able to implement the "clone" (by emulating the behaviour of a legitimate Token) on a NFC enabled mobile phone.

The second attack was to covertly read the victim's Token to retrieve the information needed to create a clone. We developed a MIDlet to emulate a contactless Reader using a standard NFC contactless communication API, JSR 257 [12]. This was relatively simple as the API surprisingly did not require a code signing certificate. The phone was then able to replay the Reader messages from the captured transaction and trick the legitimate Token to participate in a transaction.

Finally to complete the experiments both a clone Token and clone Reader were set-up on 2 separate NFC phones and they were then able to reproduce the complete original transaction from the legitimate Token and Reader. A more detailed explanation of this can be found in the published paper [9].

5 Concluding Remarks

NFC is one of the most exciting recent developments for application development and service provision via mobile phones. However, the rush towards exploiting the functionality should be balanced by the need to establish sound security measures and processes, as many transactions will involve the exchange of valuable and sensitive information. Basing security around a hardware security module (SE) is a wise step, however the uncertainty around SE architecture, ownership and management can leave cracks in the defences that can be exploited by attackers. Unusually we are not just concerned about the legitimate transactions of the NFC phone being compromised, but also the use of the phone as a skimming and clone/emulation tool. We have shown how the latter risk can become practical in a legacy NFC phone using an embedded SE. The industry seems to be moving towards the SE integrated in the SIM which has the potential to be secure. However unless this SE is made open and available to developers and third parties there is a danger that this architecture will not be incorporated into all mobile phones and applications, which leaves us back where we are today.

6 References

[1] "Third Generation Partnership Project, Specification of the Subscriber Identity Module -Mobile Equipment (SIM - ME) interface (Release 1999)", 3GPP TS 11.11 V8.14.0, June 2007.

[2] K. Finkenzeller, RFID Handbook: Radio-Frequency identification fundamentals and applications, Wiley, 1999.

[3] ISO/IEC 18092 (ECMA-340), "Information technology Telecommunications and information exchange between systems Near Field Communication Interface and Protocol (NFCIP-1)", 2004.

[4] K. Mayes, K. Markantonakis, "Mobile communications security controllers, an evaluation paper, Information Security Technical Report", Volume 13, Issue 3, August 2008, p173-192, ISSN 1363-4127.

[5] K. Mayes, K. Markantonakis, "Smart Cards, Tokens, Security and Applications", Springer Verlag, 2007.

[6] The Java Card Forum http://www.JavaCardforum.org/

[7] GlobalPlatform, Global Platform Card Specification, 2006.

[8] "Third Generation Partnership Project, Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (Release 1999)", 3GPP TS 11.14 V8.18.0, June 2007.

[9] L. Francis, G. Hancke, K. Mayes and K. Markantonakis, Potential Misuse of NFC Enabled Mobile Phones with Embedded Security Elements as Contactless Attack Platforms, in IEEE Conference Proceedings of RISC 2009 – The First International Workshop on RFID Security and Cryptography 2009, in conjunction with the 4th International Conference for Internet Technology and Secure Transactions (ICITST 2009), London, November 2009.

[10] 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.

[11] G.P. Hancke, Eavesdropping Attacks on High-Frequency RFID Tokens, Workshop on RFID Security (RFIDSec) July 2008.

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