Addressing NFC Mobile Relay Attacks: NFC User Key Confirmation Protocols

Ali Alshehri and Steve Schneider
Department of Computing, University of Surrey
Guildford GU2 7XH, England

Abstract

Near field communication (NFC) is a Radio Frequency (RF) technology that allows data to be exchanged between devices that are in close proximity. A relay attack is a general attack against RFID/NFC that manipulates the proximity factor between entities, which is caused by either internal or external attackers. We propose NFC User Key Confirmation Protocols (UKC) to address the NFC mobile relay attack. For proximity proof, the solution is intended for addressing external attackers and for the assumption that the users are honest. The main idea of addressing the relay attack in NFC mobile is by engaging the users for the key confirmation protocol. The key confirmation protocol would be a shared mission between the cryptographic protocol and the end user. The UKC protocols address both entity authentication and proximity proof requirements. We formally verify the protocol using CasperFDR.

1. Introduction

Near Field Communication (NFC) [1] is a *radio frequency* (RF) communication link, which allows data to be exchanged between devices that are normally less than 10 cm apart [2]. NFC-based devices are an emerging technology changing the way we communicate with objects. For instance, payments, tickets and coupons can be exchanged just by waving the NFC-based devices at the point of sale. NFC tends to be in mobile phones more since the majority of people already have one. NFC in mobiles can operate in three different modes determined by the application used; it can communicate with other NFC mobiles in *Peer-to-Peer mode*, communicate with a passive RFID/NFC tag in *reader/writer mode*, or communicate with an NFC reader in *card emulation mode* [3]. NFC mobile is an evolution of a passive contactless card.

The *relay attack* is a general attack against RFID/NFC that manipulates the proximity factor between entities. The attacker relays the signal between two distant parties while both, or one, of them believe they are next to each other.

There are a number of non-cryptographic measures for addressing the relay attack. *Distance bounding protocols* (DB) [4] are a group of protocols which measure the round-trip time taken to transfer a single bit between the two engaging parties. This imposes a limit based on the transmission speed of information. If there is a delay, then this suggests the time taken is more than needed. The delay indicates an attacker may be relaying the channel. DB protocols make the relay attack very difficult for the attacker and provide the best solution so far for the passive contactless cards.

However, there are some disadvantages of DB protocols. It is difficult to measure the round-trip latency caused by the different computational processes required by both parties. Therefore, both parties require additional time. As a result, the real reason for the latency would not be known, whether it is due to a computational reason or by an attacker relaying the communication. Because of the computational process problem, the DB protocols have been developed with an assumption of a constant computational time. In addition, DB protocols try to address security requirements in entity authentication beside proximity authentication. Even though entity authentication requirements are important, they are not the main requirement of the DB protocol. The DB protocols were invented firstly to address proximity, while entity authentication was included for the purpose of establishing a complete protocol which addresses both requirements. The DB protocols address more security requirements than needed, focusing on improving the less important requirements. On the other hand, it is reasonably argued that it is difficult to apply a normal entity authentication protocol alongside a proximity proof protocol (bit round-trip) because of the limited choices in a passive contactless card and the nature of the proximity proof. However, assuming a constant computational time, which breaks the core element of measuring the bit round-trip, and the focus on entity authentication issues is a major disadvantage in these protocols. In theory, the security of DB protocols can be proven mathematically, but their application in practice is questionable and requires further research.

Copyright © 2014, Infonomics Society

Another approach is by measuring the ambient conditions between parties. This is based on the fact that if the engaging parties are in close proximity, then they should share similar ambient conditions such as GPS, sound, light or temperature [5]. However, the accuracy of these measures is a major concern. A GPS signal needs an outside clear environment. Light and sound may be affected by the source's direction or interferences. Temperature can be quite similar in two far places. Not to mention if these measures require additional equipment. Nonetheless, advantages of the ambient conditions solutions are that they focus on the main problem by trying to prove the proximity and they can be implemented on top of any cryptographic protocols.

A recent approach is by utilising NFC button solutions. The NFC button is used either as a turn on/off NFC function [6, 7] or measuring the time between showing and pressing buttons showed on both devices [8]. The advantage of this solution is the user's engagement.

As far as the current relay attack measures are concerned, an ideal solution should feature the following:

1- Response time should not be a critical factor due to the difficulty associated with measuring it.
2- The ability to run on top of, or within, normal authentication protocols.
3- Easy to apply into a system.
4- Accurate.

In this paper we propose *NFC User Key Confirmation protocols* (NFC UKC). NFC UKC protocols overcome issues related to the current relay attack countermeasures by collaboration between three main elements as illustrated in Figure 1. : Cryptography, NFC mobile and the user. NFC UKC protocols, to address the relay attack, utilise a proper cryptographic measure (a *key confirmation protocol*) in conjunction with a user engagement found in the contemporarily powerful NFC mobiles.

The paper is organised as follows: Section 2 discusses different issues related to the relay attack in order to have a clear understanding of the problem and how to solve it. Section 3 demonstrates the main idea of the proposed NFC UKC protocols. Then, we illustrate the UKC protocols in the three modes of operations: card emulation mode in Section 4, peer-to-peer in Section 5, and reader/writer mode in Section 6. The discussion and residual risks are in Section 7. Finally, in Section 8 we formally verify the security of the NFC UKC protocols using CasperFDR.

2. Relay attack

Figure 2. shows the relay attack in RFID/NFC domain, in which a relay attack happens when a

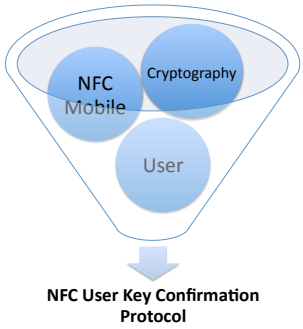


Figure 1: NFC UKC Protocol elements

contactless reader cannot distinguish between the real contactless card and the proxy-card [9]. This is applicable as well if the two parties are NFC mobiles [10, 11]. The relaying channel can be Bluetooth, Wi-Fi or Internet managed by the intruder (Mallory). Intruders in relay attacks can be external or internal. The intruder can pretend to be Bob, or Alice, if they have a relay channel of the communication between Alice and Bob, while for example Alice believes that she is communicating with Bob in close proximity. The intruder is external if all parties are honest, and an "external" attacker relays the communication while the honest parties think they are close to each other.

The broken property in a relay attack is a proximity proof property not an entity authentication property.

2.1 Relay attack and cryptographic protocols

Entity authentication protocols are used for the purpose of authenticating one entity to another. When authenticating an entity in the Internet domain, proximity is not assumed and passing messages from one router to another is normal. However, in RFID/NFC domain proximity is a critical assumption. While proximity is not a requirement in Internet protocols, it is considered a vital requirement in the RFID/NFC protocols. The reason is that a promised value given at the end of an RFID/NFC authentication protocol would be given to the one who is physically present at the reader, even if that was an intruder relaying the communication from a distant honest user, for example accessing a train gate or cashing a coupon for a free coffee. The entity authentication protocols work at distance and the analysis threat model does not include proximity. Therefore, applying a normal user authentication protocol, suitable for the Internet, is not necessarily applicable to the RFID/NFC protocols. The relay attack is an additional challenge in RFID/NFC protocols.

However, a relay attack normally occurs on authentication protocols rather than *key*

Copyright © 2014, Infonomics Society

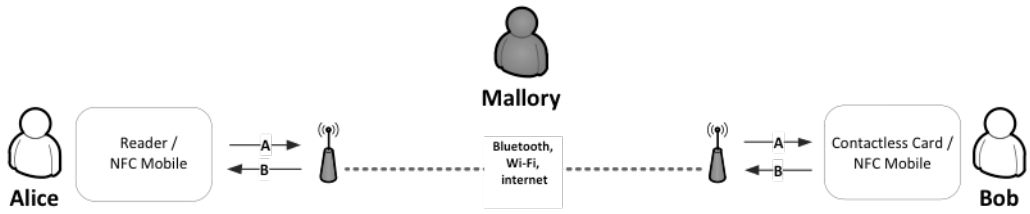


Figure 2: RFID/NFC relay attack

establishment protocols. Even if the intruder relays the communication in key establishment protocols, no benefit will be gained except if there was a flaw in the protocol itself where the intruder is able to know the new key. Key establishment is not concerned with proximity, whereas authentication can be, especially in the RFID/NFC domain.

Normally, key establishment protocols are divided into two main protocols. First, a *key generation protocol* is used to establish a fresh secret key between parties. Second, in a *key confirmation protocol* each entity has to prove that it has the correct key (e.g. encrypting a new nonce with the new key).

The solution to this is that the relay attack can be addressed by utilising the key confirmation protocol. The main idea of addressing the relay attack in NFC mobile is by engaging the users for the key confirmation protocol. The key confirmation protocol would be a shared mission between the cryptographic protocol and the end user. The mission related to the cryptographic protocol is that it is responsible for delivering a challenge, e.g. nonce, to the user encrypted under an agreed key. The mission related to the user is that the user is responsible for proving the knowledge of that challenge physically through the user themselves, which is the core idea of the NFC UKC protocols.

2.2 Relay attacks classifications

We call a relay attack *simple* when the intruder interacts with a contactless card/NFC mobile without the user's awareness. A turn on/off button for NFC function is a direct security measure for addressing such an attack, as suggested in the literature [6, 7]. The idea is to enable the NFC function only if the user explicitly activates it. In addition, the on/off button should include disabling the NFC touch-and-go feature. The NFC *touch-and-go* feature is when developing an NFC application a user could be given the choice to start the communication directly without any confirmation. Moreover, the on/off button should include disabling NFC function when the battery of the NFC device is dead.

An advanced relay attack is when the user is fully aware of a transaction and the intruder manages to do another transaction on behalf of the user.

There are two kind of the advanced relay attack, with *internal* or *external* intruders. An internal advanced relay attack occurs when one of the engaging parties is dishonest e.g. a dishonest user who collaborates with an attacker to preform a relay attack. Whereas, an external advanced relay attack occurs when all parties are honest and an external attacker who tries to relay the communication.

The proposed solution, the NFC UKC protocols, addresses the internal advanced relay attack, and intended for addressing external attackers with the assumption that the users are honest.

3. NFC User Key Confirmation Protocols

NFC mobiles have revolutionised the concept of contactless cards. NFC mobiles are more powerful than just passive contactless cards in terms of two additional features: operating in different modes and opening a channel to interact with the user. In fact, one requires mobiles to show in practice that a key confirmation protocol is better than an entity authentication protocol for addressing the relay attack. The reason is that the key confirmation protocol requires a response by the user through the contactless card, which becomes possible in NFC mobiles.

3.1 Terminology

- **NFC mobile**: an NFC mobile can be a smart phone or a tablet with an NFC technology.
- **Reader**: a reader is a contactless reader supported with a touch screen and has a relationship with users, e.g. a train gate or a coffee cashier.
- **Contactless card/ Tag**: is a passive object only activated by the power of other NFC devises, stores some information and able to preform basic encryption/decryption, e.g. contactless credit card or a tag in a poster issuing coupons.

Copyright © 2014, Infonomics Society

- **Users**: a user is a person who carries the NFC mobile. The participation of a user in NFC UKC protocols depends on the NFC device used but at least one is necessarily. We assume the user does not necessarily exist behind an NFC reader, and it is not applicable with a contactless card since the user can not interact through a contactless card.
- **Prover**: as a protocol role, an entity is a Prover when it is required to prove its proximity.
- **Verifier**: as a protocol role, an entity is a Verifier when it verifies a proximity authentication of a Prover.

3.2 Proximity authentication methods

There are two main methods in which a device can prove its proximity in the UKC protocols, *Proximity Token* and *Proximity Challenge/Response.* The user can prove to other NFC devices that their NFC mobile is in a close distance by performing Proximity Challenge/Response with other NFC devices. On the other hand, the user an verify the proximity of other NFC devices by making an informed decision on whether the other NFC device is in close distance by an entity's Proximity Token. The user and the NFC mobile together share the proving/verifying process. Both methods are explained in the following sections.

3.2.1 Proximity Token

As illustrated in Figure 3a, this method is for the reader and the contactless card (Prover) to prove their proximity to the user who carries the NFC mobile (Verifiers). The Prover sends its Proximity Token (PrxTok) to the user, through the user's NFC mobile. The verification step is shared between the NFC mobile and the user, where the NFC mobile manages the encryption/decryption of the PrxTok between the user and other devices, and the user is the one who makes a decision of the PrxTok's authenticity. The PrxTok is a message that includes information which enables the user to verify the proximity of the Prover, including the following:

- Information about the Prover. The aim is that the Prover tells the user about itself so that the user can make a decision whether the description of the Prover is the same one that the user is dealing with. Prover's information may include location, name, a photo from the user's point of view (the shop, the cashier or the point of sale), the name of the employer, etc.
- Information about the user (the Verifier). The Prover tells the user what it believes about the user, so that the user can make sure it is the one who the Prover believes it is. Information includes name, location, etc. If information about the NFC mobile were included as well, this would enhance the security more.
- Information about the process that is being done between the Prover and the user such as price, transaction details, etc.

The security of the PrxTok is vital. External intruder cannot know the PrxTok even with relaying the communication because information is encrypted and decrypted at the application layer and confirmed at the user's level. If the intruder relays the PrxTok, the user would be able to tell the contradictions in the PrxTok(s). In addition, it is very difficult to create such information because this information is either variable (e.g. the name of the employer, current date and amount of the transaction) or static that it would be detected by the user if faked (e.g. cashier location). Even if one assumes an intruder manages to fake a PrxTok, it would be impossible to send it without knowing the shared key as the PrxTok is always encrypted.

3.2.2 Proximity Challenge/Response

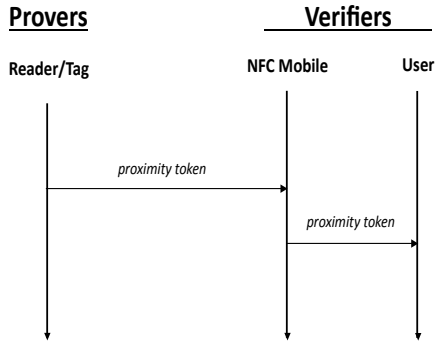
As illustrated in Figure 3b, the other approach is for the user with an NFC mobile (Provers) to physically prove their proximity to other devices (Verifier). Here, the Verifier is a reader or another NFC mobile. The user with the NFC mobile can prove the proximity to the reader and another NFC mobile but not a contactless card because the user's engagement is vital. Proximity Challenge/Response method includes the following steps:

1. Verifier generates a proximity challenge and sends it to the user's mobile encrypted with a shared key.
2. The user's mobile shows the proximity challenge to the user by decrypting the proximity challenge with the shared key.
3. The user themselves proves to the Verifier the knowledge of the proximity challenge by a correct proximity response.

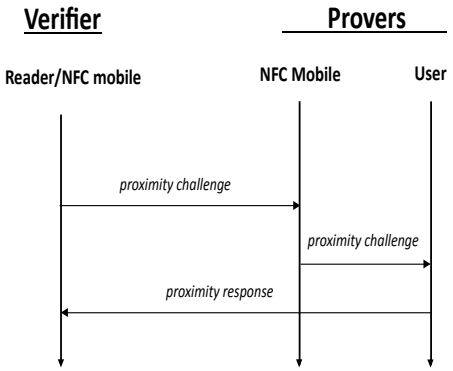
The nature of the proximity challenge takes different forms depending on the device used such as picture or word. However, proximity response must be done through the user.

The random picture/word can only be known after the user interacts with the prover. Knowing the picture/word after finishing the protocol is harmless as one uses the random picture/word for a one time authentication not secrecy. Moreover, the user will not start interacting with a prover, which may be an external intruder, except if the PrxTok is approved.

Copyright © 2014, Infonomics Society



(a) Proximity token



(b) Proximity challenge/response

Figure 3: Methods of proximity authentication in the NFC User Key Confirmation protocols

3.3 Security characteristics

Table 1. shows the security characteristics of the NFC UKC protocols. The NFC UKC in all three modes of operations provide both mutual entity authentications and mutual proximity proofs, except for the UKC writer/reader mode where only unilateral proximity proof is provided. Entity authentication addresses both internal and external intruders. Proximity proof addresses only external intruders, and controlling internal attacks can be done through the system policy, for example a penalty or prevention from using the system.

Finally, a key generation protocol is required for the NFC UKC protocols. Entities must either share a key before the beginning of the UKC protocol, or they have just gone through a key generation protocol resulting in a new fresh key. The NFC UKC protocols are illustrated with the use of long term secret keys shared between the entities.

4. NFC UKC protocol in card emulation mode

In this mode, NFC mobile emulates a contactless card and interacts with a reader. For example, mobile wallet (credit card) and train gate applications. We propose a mutual proximity proof and mutual entity authentication in the NFC UKC card emulation protocol.

Table 2. illustrates protocol notations. The protocol is as follows (messages in **bold** occur at the protocol layer, other messages occur in the physical world):

1. Reader → User's mobile : {Reader, User, Reader PrxTok, Reader rPe} $K_{Reader/User's\ mobile}$
2. User's mobile → User : Reader PrxTok, Reader rPe
3. Reader → User : display 4 pictures
4. User → Reader : select Reader rPe

The shared key $K_{Reader/User's\ mobile}$ is a key shared between the reader and the user's mobile. In message 1, the reader encrypts the following:

- Both identities: Reader and User.
- Reader rPe: A random picture (a proximity challenge).
- Reader PrxTok: A PrxTok method from the reader (includes all three aspect: information about prover, verifier and transaction).

Message 1, the Reader PrxTok is used by the user to authenticate the reader. The user makes a decision whether the reader is in close proximity based on the Reader PrxTok. The user can see the random picture Reader rPe, say an apple, on the mobile screen by decrypting the picture with the shared key (message 2). Message 3, the reader displays the Reader rPe among other choices, say (banana, orange and grape), in order to examine whether the user is able to choose the right picture. Finally, the user chooses the right picture from four different pictures on a touch screen placed on the reader. The Reader rPe is like a nonce but in addition it has a meaning which can be verified and understood by the user who has just seen it. If the user chooses the right picture, then this indicates for the reader that the user is in close proximity.

Copyright © 2014, Infonomics Society

Table 1: UKC protocols security characteristics

	Entity authentication				Proximity proof			
	Unilateral	Mutual	Intruder		Unilateral	Mutual	Intruder	
			Internal	External			Internal	External
UKC card emulation		✓	✓	✓		✓	✗	✓
UKC peer-to-peer		✓	✓	✓		✓	✗	✓
UKC writer/reader		✓	✓	✓	✓		✗	✓

For proximity proof, the reader proves its proximity by a PrxTok method. The NFC mobile with the user proves their proximity by a proximity challenge/response method. The challenge is a random picture rPe sent to the NFC mobile by the reader, and the response is that the user chooses the right picture from a set of pictures at a touch screen placed on the reader.

For entity authentication, the reader is authenticated to the user through the PrxTok and the shared key. The information included in the PrxTok can be utilised as a time stamp and a random number for authenticating the reader. The user is authenticated by the proximity challenge/response method.

5. NFC UKC protocol in Peer-To-Peer mode

In NFC peer-to-peer mode, two NFC mobiles are interacting with each other. We propose a mutual proximity proof and a mutual entity authentication in the NFC UKC peer-to-peer. An example of such application is NFC mobile-to-mobile money transactions.

Each user with an NFC mobile can verify the proximity of another NFC mobile by checking the PrxTok. The proximity challenge/response method is used for a mutual entity authentication. Each user with an NFC mobile performs a proximity challenge/response with another NFC mobile. The challenge is a random word rWd chosen by the user.

The protocol is as follows (messages in **bold** occur at the protocol layer, other messages occur in the physical world):

1. Alice → Alice's mobile : Alice rWd
2. Bob → Bob's mobile : Bob rWd
3. Alice's mobile → Bob's mobile : {Alice, Bob, Alice PrxTok, Alice rWd} $K_{Alice's\ mobile/Bob's\ mobile}$
4. Bob's mobile → Alice's mobile : {Bob, Alice, Bob PrxTok, Bob rWd} $K_{Alice's\ mobile/Bob's\ mobile}$
5. Alice's mobile → Alice : Bob PrxTok, Bob rWd
6. Bob's mobile → Bob : Alice PrxTok, Alice rWd
7. Alice → Bob : Bob rWd
8. Bob → Alice : Alice rWd

The shared key $K_{Alice's\ mobile/Bob's\ mobile}$ is a key shared between Alice's mobile and Bob's mobile. In message 1, Alice types a word on her mobile and Bob does the same in message 2. In message 3, Alice's mobile sends (Alice, Bob, Alice PrxTok, Alice rWd) to Bob encrypted with the shared key $K_{Alice's\ mobile/Bob's\ mobile}$, and Bob does the same in message 4. In messages 3 and 4, random words are decrypted to Alice and Bob through their NFC mobiles, and both of them can verify the other entity's proximity through the PrxToks. Finally, in message 7 Bob asks Alice to verbally verify the random word that he has sent, and Alice does the same in message 8 where a mutual entity authentication is done.

The random word can be a random picture rPe where in message 1 and 2 both users choose a picture from a set of pictures on their mobile screens. Then, in message 7 and 8 both users confirm verbally the random picture received.

6. NFC UKC protocol Reader/Writer mode

In NFC reader/writer mode, the user who has a NFC mobile performs the protocol with a passive card. Even though the protocol achieves a mutual entity authentication, only the NFC mobile usually proves its proximity to the passive contactless card because it is difficult for the user to interact with a contactless card or tag. The contactless card/tag proves its proximity to the user by PrxTok method. A proximity challenge/response method is used for freshness and enhancing the security.

The protocol is as follows (messages in **bold** occur at the protocol layer, other messages occur in the physical world):

1. User's mobile → User : display 4 pictures.
2- User → User's mobile : select User rPe.
3- User's mobile → Tag : {User, Tag, User PrxTok , User rPe} $K_{User's\ mobile/Tag}$
4. Tag → User's mobile : {User, Tag, Tag PrxTok, User PrxTok , User rPe} $K_{User's\ mobile/Tag}$
5. User's mobile → User: Tag PrxTok, User PrxTok ,User rPe.

The shared key $K_{User's\ mobile/Tag}$ is a key shared between the user's mobile and the tag. In message 1, the user's mobile displays four pictures to the user, and the user chooses one of them User rPe (message 2). In message 3 the user's mobile encrypts to the tag

Copyright © 2014, Infonomics Society

Table 2: Protocol notation

Reader	A contactless reader such as a train gate.
User's mobile	A mobile with NFC technology.
User, Alice and Bob	The person who hold a NFC mobile.
PrxTok	Proximity Token method.
rPe	A random picture
rWd	A random word.
K	A shared key.

the following: User, Tag, User PrxTok and User rPe . Here, the User PrxTok is used for authenticating the user to the tag, but not for proving the user's proximity. In message 4, the tag encrypts the received (User, Tag , User PrxTok , User rPe) with the Tag PrxTok which is stored in the tag memory. In message 5, the user checks Tag PrxTok, User PrxTok that it similar to the one they sent and the User rPe.

7. Discussion and Residual risks

A main difference of the UKC protocols from other relay attack countermeasures is that this solution is a collaboration between the cryptographic protocol and the user themself for transferring the proximity proof. It could be argued that user engagement is a disadvantage. However, an NFC application has a different assumption from Internet communication where the users are engaged when presenting their mobile and show their intention by touching the other NFC device. In fact, user engagement in the authentication process is one of the laws of identity for successful, proper identification [14].

In a real world application, information included inside a PrxTok should consider the usability factor by selecting some information rather than showing all the information.

Guessing is still a residual risk. There is a chance for the intruder to guess the random picture presented on the reader. A solution is to increase the number of pictures and to select permutation. For example, to present 6 pictures with two random pictures to be chosen in sequence. In this case, the possibility for the intruder to guess to be right would be very low (first correct guess is 1/6, followed by second guess correct at 1/5, gives a total of 1/6 * 1/5 = 1/30 approximately 3%).

Proximity proof, in the current development of the NFC UKC protocols, does not hold with dishonest users (internal intruders) because a distant honest user can tell a close attacker the right picture, which can be addressed by the system policy.

8. Modelling and analysis of NFC User Key Confirmation

8.1 CasperFDR

In our analysis we use *Communicating Sequential Processes* CSP [15], with its model checker *Failures Divergence Refinement* (FDR2), which is proven to be an effective method in analysing the security of protocols [16]. However, modelling protocols in CSP is not a trivial task. Gavin Lowe developed *CasperFDR* [17], a tool that allows the user to write an abstract description of a security protocol, then the tool produces a model in the CSP language, and directly checks it with FDR2. CasperFDR has been used to analyse a huge number of protocols [18, 19], which proves its capability of finding vulnerabilities.

CasperFDR is a formal method tool which supports symbolic protocol analysis in the Dolev-Yao model [13] which assumes that no encrypted message can be decrypted without the decryption key, thus the CasperFDR intruder model does not perform any cryptanalysis. However, the intruder does have full control of the network traffic, and tries to break the security protocol from what passes on the network.

CasperFDR performs a refinement check of the protocol against its requirements. When refinement fails, then it provides a trace which shows how the property fails, that corresponds to an attack. Moreover, CasperFDR manages the Xor operation where attacks against these algebraic properties are considered in CasperFDR.

CasperFDR allows customisation of the intruder's ability to access or interfere with specific messages of the protocol, where the messages can have any combination of the following annotations:

- **C:** (Confidential) The intruder cannot eavesdrop this message.
- **NF:** No fake by the intruder. This means that the intruder cannot fake this message: any message received must have been generated by the claimed sender.
- NRA No reascribing (changing the sender ID), any message received must have the sender ID that was issued.
- NRA- No honest reascribing (changing the sender ID except to a dishonest user ID, of an eavesdropped message)

Copyright © 2014, Infonomics Society

- **NR** No redirecting (changing the receiver ID), wherever this message is sent then it cannot arrive at the wrong destination.
- **NR-** No honest redirecting (changing the receiver ID except to his own ID, of an eavesdropped message).

The advantages of CasperFDR for the modelling of NFC protocols fall into the following aspects:

1- It allows modifying the intruder's power on every channel of the protocol. Such settings are needed to capture some behaviour in the analysed protocols. This is not possible in AVISPA and Scyther, while in CasperFDR it is a default setting.
2- Accessing the original CSP code which allows direct modification of the model when appropriate.
3- The ability to use the tool for more advanced analysis such as capturing various security requirements in NFC mobile coupon. This is also possible in AVISPA and Scyther.
4- It features various automated and robust security specifications. This is also provided in AVISPA and Scyther but not as powerful as in CasperFDR.

8.2 Formal modelling by CasperFDR

The NFC User Key Confirmation protocols are modelled in all three modes of operations; see Appendixes A.1, A.2 and A.3. The entity authentication properties are formally verified by Casper, but not the proximity proof properties since Casper cannot capture proximity between entities.

8.2.1 Modelling the user and their mobile

There are some messages in the model between the user and their mobile. It is unlikely for a dishonest entity to access this range of the communication except if malicious software is downloaded. The intruder is assumed to have no control over the range between the user and his or her mobile. We conceal the communication between the user and their mobile from the intruder. In order to model this channel, we adjust the Channel section to annotate this channel as follows:

```
#Channels
2 C NF NRA NR
```

The second line means that in message 2 the intruder neither can eavesdrop C, nor fake data NF, nor reascribe NRA or redirect NR. We apply the channel setting for the card emulation mode model in message (2) Appendix A.1, the peer-to-peer mode in messages (1,2,5,6) Appendix A.2 and the reader/writer mode in messages (1,4) Appendix A.3.

8.2.2 Modelling the reader's touch screen

Another aspect of the model is capturing the act of the reader by presenting four pictures to the user who chooses the right one. We model this as follows:

```
#Protocol description
0.              -> reader : nfcMobile
1. reader -> nfcMobile : {reader, user, prxtok,   rPe}{kab}
2. nfcMobile -> user : prxtok, rPe
3. -> reader : user
4. user -> reader : rPe

#Channels
2 C NF NRA NR
```

The intruder is assumed to have no control over message 2 because it occurs between the user and their mobile. The reader starts a communication with the user at message 3. Then, the user provides the right picture to the reader.

In order to examine this model, we run the protocol with 3 different users. CasperFDR finds no attack. In addition, it is possible now to check if the intruder is able to know the picture (which we expect they should) by checking this specification:

```
#Specification
StrongSecret(reader, rPe, [user])
```

As expected, the model fails this specification: the result is an interaction with which the intruder is able to learn the picture by the end of the protocol run. This does not affect the authentication between the reader and the user. The knowledge of the picture after it has been used for authentication is harmless as it is randomly generated. Secrecy of the picture is not a requirement. The requirement is authentication.

Hence, we formally verified the security of these protocols and no attacks were found.

9 Conclusion

We proposed the NFC User Key Confirmation protocols to address the relay attack in NFC. The UKC protocols are collaboration between the cryptographic protocols, the user and the NFC mobile in an effort to prove proximity. We illustrated the UKC protocols in the three NFC modes of operations (card emulation, peer-to-peer and writer/reader).

The NFC UKC in all three modes of operations provide both mutual entity authentications and mutual proximity proofs, except for the UKC writer/reader mode where only unilateral proximity proof is provided. Entity authentication addresses both internal and external intruders. However, proximity proof addresses only external intruders,

Copyright © 2014, Infonomics Society

and controlling internal attacks can be done through the system policy, for example a penalty or prevention from using the system.

There are a number of advantages of our solution. In contrast to Distance bounding protocols, the response time is not a critical factor in our solution, and it is easy to apply with any existing entity authentication protocols. In addition, It is more accurate than measuring the ambient condition approaches.

We formally verified the entity authentication properties, but not the proximity proof, in our solution by CasperFDR. Future works are to apply NFC UKC protocol with public keys, considering the usability aspect and enhancing the solution to address the internal intruders.

10. References

[1] ISO/IEC, "Information technology – telecommunications and information exchange between systems – near field communication – interface and protocol (NFCIP- 1)," 2004.

[2] K. Finkenzeller, *RFID Handbuch: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, 3rd ed. John Wiley and Sons, Ltd., 2010.

[3] V. Coskun, K. Ok, and B. Ozdenizci, *Near Field Communication: from theory to practice*, 1st ed. JohnWiley and Sons, Ltd, 2012.

[4] S. Brands and D. Chaum, "Distance-bounding protocols," in *Advances in Cryptology EUROCRYPT 93*, ser. Lecture Notes in Computer Science, T. Helleseth, Ed. Springer Berlin Heidelberg, 1994, vol. 765, pp. 344–359. [Online]. Available: http://dx.doi.org/10.1007/3-540-48285-7 30.

[5] P. Urien and S. Piramuthu, "Identity-based authentication to address relay attacks in temperature sensor-enabled smartcards," in *Smart Objects, Systems and Technologies (SmartSysTech), Proceedings of 2013 European Conference on*, June 2013, pp. 1–7.

[6] G. Madlmayr, J. Langer, C. Kantner, and J. Scharinger, "Nfc devices: Security and privacy," *in Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, 2008, pp. 642–647.

[7] A. Alshehri, J. A. Briffa, S. Schneider, and S. Wesemeyer, "Formal security analysis of nfc m-coupon protocols using casper/fdr," *in Near Field Communication (NFC), 2013 5th International Workshop on*, 2013, pp. 1–6.

[8] S. Kang, J. Kim, and M. Hong, "Button-based method for the prevention of near field communication relay attacks," *International Journal of Communication Systems*, pp. n/a–n/a, 2014. [Online]. Available: http://dx.doi.org/10.1002/dac.2751

[9] G. P. Hancke, "A practical relay attack on iso 14443 proximity cards," *Technical report, University of Cambridge Computer Laboratory*, pp. 1–13, 2005.

[10] L. Francis, G. P. Hancke, K. Mayes, and K. Markantonakis, "Practical NFC peer-to-peer relay attack using mobile phones," in *RFIDSec*, 2010, pp. 35–49.

[11] M. Roland, J. Langer, and J. Scharinger, "Applying relay attacks to google wallet," *in Near Field Communication (NFC), 2013 5th International Workshop on*, Feb 2013, pp. 1–6.

[12] C. Cremers, K. Rasmussen, B. Schmidt, and S. Capkun, "Distance hijacking attacks on distance bounding protocols," in *Security and Privacy (SP), 2012 IEEE Symposium on*, May 2012, pp. 113–127.

[13] D. Dolev and A. Yao, "On the security of public-key protocols," *IEEE Transactions on Information Theory*, vol. 2, no. 29, 1983.

[14] M. C. Kim Cameron, *The Laws of Identity*, 2005. [Online]. Available: msdn.microsoft.com

[15] C. A. R. Hoare, *Communicating Sequential Processes*. Prentice-Hall, 1985.

[16] P. Y. A. Ryan, S. A. Schneider, M. Goldsmith, G. Lowe, and A. W. Roscoe, *Modelling and analysis of security protocols*. Addison-Wesley-Longman, 2001.

[17] G. Lowe, "Casper: A compiler for the analysis of security protocols," *Journal of Computer Security*, vol. 6, no. 1-2, pp. 53–84, 1998.

[18] B. Donovan, P. Norris, and G. Lowe, "Analyzing a library of security protocols using Casper and FDR," in *Proceedings of the Workshop on Formal Methods and Security Protocols*, 1999, some of the Casper scripts are available here: http://web.comlab.ox.ac.uk/oucl/work/gavin.lowe/Security/Papers/prots.tar.gz.

[19] S. Abughazalah, K. Markantonakis, and K. Mayes, "A mutual authentication protocol for low-cost rfid tags formally verified using casperfdr and avispa," in *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for*, Dec 2013, pp. 44–51.

11. Acknowledgements

Thanks to the Ministry of Higher Education in Saudi Arabia for supporting this research.

A. Appendix

A.1 Casper Modelling of NFC UKC Card Emulation protocol

```
#Free variables
reader, user, nfcMobile  : Agent
prxtok : PrxToks
rPe  : RandomPicture
kab : SessionKey
```

Copyright © 2014, Infonomics Society

```
InverseKeys = (kab,kab)

#Processes
INITIATOR(reader,user,kab,prxtok, rPe)
RESPONDER(nfcMobile,user,kab)
RESPONDER2(user,reader,nfcMobile,prxtok)

#Protocol description
0.      -> reader : nfcMobile
1.  reader -> nfcMobile : {reader,user, prxtok, rPe}{kab}
2.  nfcMobile -> user : prxtok, rPe
3.      -> reader : user
4.  user -> reader : rPe

#Channels
2  C NF NRA NR

#Specification
Agreement(user, reader, [rPe])
Agreement(reader, user, [prxtok])

#Actual variables
Reader, User, NfcMobile, I ,User2 ,NfcMobile2,User3
,NfcMobile3: Agent
Prxtok,MPrxtok  : PrxToks
RPe , RPe2, RPe3,MRPe : RandomPicture
Kab : SessionKey
InverseKeys = (Kab,Kab)

#System
INITIATOR(Reader,User,Kab,Prxtok, RPe)
INITIATOR(Reader,User2,Kab,Prxtok, RPe2)
INITIATOR(Reader,User3,Kab,Prxtok, RPe3)
RESPONDER(NfcMobile,User,Kab)
RESPONDER(NfcMobile2,User2,Kab)
RESPONDER(NfcMobile3,User2,Kab)
RESPONDER2(User,Reader,NfcMobile,Prxtok)
RESPONDER2(User2,Reader,NfcMobile2,Prxtok)
RESPONDER2(User3,Reader,NfcMobile2,Prxtok)

#Intruder Information
Intruder = I
IntruderKnowledge = {Reader, User,NfcMobile, I, MPrxtok,
MRPe, User2,NfcMobile2,User3 ,NfcMobile3}
```

A.2 Casper Modelling of NFC UKC peer-to-peer protocol

```
#Free variables
alice, bob, aliceMobile, bobMobile : Agent
prxtokAlice, prxtokBob : PrxToks
aliceRWd, bobRWd : RandomWorlds
kab : SessionKey
InverseKeys = (kab,kab)

#Processes
INITIATOR(alice,bob,aliceMobile,aliceRWd,prxtokBob,prxto
kAlice)
RESPONDER(bob,alice,bobMobile,bobRWd,
prxtokAlice,prxtokBob)
RESPONDER2(aliceMobile,alice,bob, bobMobile,
kab,prxtokAlice)
RESPONDER3(bobMobile,bob,alice,aliceMobile,kab,prxtok
Bob)

#Protocol description
0.      -> alice          : aliceMobile
1.  alice -> aliceMobile     : aliceRWd
2.  bob -> bobMobile         : bobRWd
3.  aliceMobile -> bobMobile : {alice, bob, prxtokAlice,
aliceRWd}{kab}
4.  bobMobile -> aliceMobile : {bob, alice, prxtokBob,
bobRWd}{kab}
5.  aliceMobile -> alice     : prxtokBob, bobRWd
6.  bobMobile -> bob         : prxtokAlice, aliceRWd
7.  alice -> bob             : bobRWd
8.  bob -> alice             : aliceRWd

#Channels
1  C NF NRA NR
2  C NF NRA NR
5  C NF NRA NR
6  C NF NRA NR

#Specification
Agreement(bob, alice, [prxtokBob,aliceRWd])
Agreement(alice, bob, [prxtokAlice,bobRWd])

#Actual variables
Alice, Bob, AliceMobile, BobMobile, I : Agent
PrxtokAlice, PrxtokBob , PrxtokI : PrxToks
AliceRWd, BobRWd, IRWd : RandomWorlds
Kab : SessionKey
InverseKeys = (Kab,Kab)

#System
INITIATOR(Alice,Bob,AliceMobile,AliceRWd,PrxtokBob,Prxt
okAlice)
RESPONDER(Bob,Alice,BobMobile,BobRWd,
PrxtokAlice,PrxtokBob)
RESPONDER2(AliceMobile,Alice,Bob, BobMobile,
Kab,PrxtokAlice)
RESPONDER3(BobMobile,Bob,Alice,AliceMobile,Kab,Prxto
kBob)

#Intruder Information
Intruder = I
IntruderKnowledge = {Alice, Bob, AliceMobile, BobMobile,
I,IRWd,PrxtokI}
```

A.3 Casper Modelling of NFC UKC reader/writer protocol

```
#Free variables
tag, user, nfcMobile  : Agent
prxtokUser, prxtokTag : PrxToks
rPe : RandomPicture
kab : SessionKey
InverseKeys = (kab,kab)

#Processes
INITIATOR(user,tag,nfcMobile,prxtokUser, prxtokTag,rPe)
RESPONDER(nfcMobile,user,tag, kab,prxtokUser)
RESPONDER2(tag,nfcMobile,kab,prxtokTag)

#Protocol description
0.      -> user : nfcMobile
1.  user -> nfcMobile : rPe
2.  nfcMobile -> tag : {tag,user, prxtokUser, rPe}{kab}
3.  tag -> nfcMobile : {tag,user, prxtokTag, rPe}{kab}
4.  nfcMobile -> user :  rPe, prxtokTag

#Channels
1  C NF NRA NR
4  C NF NRA NR

#Specification
Agreement(tag, user, [rPe, prxtokTag])
Agreement(user, tag, [prxtokUser])

#Actual variables
Tag, User, NfcMobile, I  : Agent
PrxtokUser, PrxtokTag, PrxtokI : PrxToks
RPe,RPeI  : RandomPicture
```

Copyright © 2014, Infonomics Society

```
Kab : SessionKey
InverseKeys = (Kab,Kab)

#System
INITIATOR(User,Tag,NfcMobile,PrxtokUser,
PrxtokTag,RPe)
RESPONDER(NfcMobile,User,Tag, Kab,PrxtokUser)
RESPONDER2(Tag,NfcMobile,Kab,PrxtokTag)
#Intruder Information
Intruder = I
IntruderKnowledge = {Tag, User,NfcMobile, I,PrxtokI,RPeI}
```

Copyright © 2014, Infonomics Society