# Security in Near Field Communication (NFC)

*Strengths and Weaknesses*

Ernst Haselsteiner and Klemens Breitfuß

Philips Semiconductors
Mikronweg 1, 8101 Gratkorn, Austria
ernst.haselsteiner@philips.com
klemens.breitfuss@philips.com

**Abstract.** This paper gives a comprehensive analysis of security with respect to NFC. It is not limited to a certain application of NFC, but it uses a systematic approach to analyze the various aspects of security whenever an NFC interface is used. The authors want to clear up many misconceptions about security and NFC in various applications. The paper lists the threats, which are applicable to NFC, and describes solutions to protect against these threats. All of this is given in the context of currently available NFC hardware, NFC applications and possible future developments of NFC.

## 1 Introduction

NFC stands for Near Field Communication. The specification details of NFC can be found in ISO 18092 [1]. The main characteristic of NFC is that it is a wireless communication interface with a working distance limited to about 10 cm. The interface can operate in several modes. The modes are distinguished whether a device creates its own RF field or whether a device retrieves the power from the RF field generated by another device. If the device generates its own field it is called an active device, otherwise it is called a passive device. Active devices usually have a power supply, passive devices usually don't (e.g. contactless Smart Card). When two devices communicate three different configurations are possible. These are described in Table 1:.

**Table 1:** Communication Configurations

| Device A | Device B | Description |
|---|---|---|
| Active | Active | When a device sends data it generates an RF field. When waiting for data a device does not generate an RF field. Thus, the RF field is alternately generated by Device A and Device B |
| Active | Passive | The RF field is generated by Device A only |
| Passive | Active | The RF filed is generated by Device B only |

These configurations are important because the way data is transmitted depends on whether the transmitting device is in active or passive mode.

In active mode the data is sent using amplitude shift keying (ASK) [1],[2]. This means the base RF signal (13,56 MHz) is modulated with the data according to a coding scheme. If the baudrate is 106 kBaud, the coding scheme is the so-called modified Miller coding. If the baudrate is greater than 106 kBaud the Manchester coding scheme is applied. In both coding schemes a single data bit is sent in a fixed time slot. This time slot is divided into two halves, called half bits. In Miller coding a zero is encoded with a pause in the first half bit and no pause in the second half bit. A one is encoded with no pause in the first bit, but a pause in the second half bit. In the modified Miller coding some additional rules are applied on the coding of zeros. In the case of a one followed by a zero, two subsequent half bits would have a pause. Modified Miller coding avoids this by encoding a zero, which directly follows a one with two half bits with no pause.

In the Manchester coding the situation is nearly the same, but instead of having a pause in the first or second half bit, the whole half bit is either a pause or modulated. Besides the coding scheme also the strength of the modulation depends on the baudrate.

For 106 kBaud 100% modulation is used. This means that in a pause the RF signal is actually zero. No RF signal is sent in a pause. For baudrates greater than 106 kBaud 10% modulation ratio is used. According to the definition of this modulation ratio [1], this means that in a pause the RF signal is not zero, but it is about 82% of the level of a non paused signal. This difference in the modulation strength is very important from a security point of view as we will describe later on in the security analysis.

In passive mode the data is sent using a weak load modulation. The data is always encoded using Manchester coding with a modulation of 10%. For 106 kBaud a subcarrier frequency is used for the modulation, for baudrates greater than 106 kBaud the base RF signal at 13.56 MHz is modulated.

Additionally to the active and passive mode, there are two different roles a device can play in NFC communication. NFC is based on a message and reply concept. This means one device A sends a message to another device B and device B sends back a reply. It is not possible for device B to send any data to device A without first receiving some message from device A, to which it could reply. The role of the device A which starts the data exchange is called initiator, the role of the other device is called target. The following Table 2: lists all possible combinations of this role with respect to the active or passive mode.Only the combination Initiator and Passive is not possible.

**Table 2:** Possible Combinations Active/Passive with Initiator/Target

| | **Initiator** | **Target** |
|---|---|---|
| **Active** | Possible | Possible |
| **Passive** | Not Possible | Possible |

Furthermore it should be mentioned that NFC communication is not limited to a pair of two devices. In fact one initiator device can talk to multiple target devices. In this case all target devices are enabled at the same time, but before sending a message, the initiator device must select a receiving device. The message must then be ignored by all non selected target devices. Only the selected target device is allowed to answer to the received data. Therefore, it is not possible to send data to more than one device at the same time (i.e. broadcasting messages are not possible).

## 2 Applications

It is impossible to give a complete picture of NFC applications as NFC is just an interface. The following sub sections introduce three example applications. These shall be viewed as typical use cases and where chosen to motivate the list of possible threats given in the next section.

### 2.1 Contactless Token

This covers all applications, which use NFC to retrieve some data from a passive token. The passive token could be a contactless Smart Card, an RFID label, or a key fob. Also, the token could be physically included in a device without any electric connections to that device.

What is important is that the only interface of the token is the contactless interface. This means it cannot act as a communication link to a device main CPU of a device because it cannot connect to the device main CPU via a contact interface. Let us also assume that the token has rather limited computing power, so it cannot run any complex protocols. The primary use would be to store some data, which can then conveniently be read by an active NFC device. Examples of such data would be a URL stored in a tag of a consumer product or the user guide of such a product. The user could then read the tag and get automatically linked to the support web page of that product. A different example would be to store the configuration data needed to access a WiFi network. New users could then easily configure their laptops to be connected to the network.

### 2.2 Ticketing / Micro Payment

in this example application, the NFC interface is used to transfer some valuable information. The ticket or the micro payment data is stored in a secure device. This could be a contactless Smart Card, but could as well be a mobile phone. When the user wants to perform a payment or use the stored ticket, the user presents the device to a reader, which checks the received information and processes the payment or accepts/rejects the ticket.

In this application example the user device must be able to perform a certain protocol with the reader. A simple read operation will not be sufficient in most cases. Also, the user device is likely to have a second interface which is used to load money or to buy tickets. This second interface can for example be linked to the mobile phone CPU. The ticket data could then be loaded into the mobile phone via the cellular network.

In this application sometimes the term 'Secure NFC' is used. However, this does not at all mean that the NFC link is somehow secured. In fact the name is rather misleading. The name just denotes a configuration using an NFC hardware chip in combination with a Smart Card chip. It should be called 'Secure Smart Card and NFC', but unfortunately the shorter name is used quite often.

### 2.3 Device Pairing

In this application the two devices communicating would belong to the same group of devices. An example could be a laptop and a digital camera. The user wants to establish a Bluetooth connection between the two devices to exchange image data. The Bluetooth link is established by bringing the two devices close together and running a given protocol over NFC between the two devices. This makes it obvious for the user which two devices get actually linked and takes away the burden of navigating through menus and selecting the right devices from lists of possible communication partners.

It should be noted that the NFC connection itself in this example is only used to establish the Bluetooth link. Image data is not transferred over NFC because NFC's bandwidth is simply too small for transferring big amounts of data.

## 3 Threats

### 3.1 Eavesdropping

Because NFC is a wireless communication interface it is obvious that eavesdropping is an important issue. When two devices communicate via NFC they use RF waves to talk to each other. An attacker can of course use an antenna to also receive the transmitted signals. Either by experimenting or by literature research the attacker can have the required knowledge on how to extract the transmitted data out of the received RF signal. Also the equipment required to receive the RF signal as well as the equipment to decode the RF signal must be assumed to be available to an attacker as there is no special equipment necessary.

The NFC communication is usually done between two devices in close proximity. This means they are not more than 10 cm (typically less) away from each other. The main question is how close an attacker needs to be to be able to retrieve a usable RF signal. Unfortunately, there is no correct answer to this question. The reason for that is the huge number of parameters which determine the answer. For example the distance depends on the following parameters, and there are many more.

- RF filed characteristic of the given sender device (i.e. antenna geometry, shielding effect of the case, the PCB, the environment)
- Characteristic of the attacker's antenna (i.e. antenna geometry, possibility to change the position in all 3 dimensions)
- Quality of the attacker's receiver
- Quality of the attacker's RF signal decoder
- Setup of the location where the attack is performed (e.g. barriers like walls or metal, noise floor level)
- Power sent out by the NFC device

Therefore any exact number given would only be valid for a certain set of the above given parameters and cannot be used to derive general security guidelines.

Additionally, it is of major importance in which mode the sender of the data is operating. This means whether the sender is generating it's own RF field (active mode) or whether the sender is using the RF field generated by another device (passive mode). Both cases use a different way of transmitting the data and it is much harder to eavesdrop on devices sending data in passive mode.

In order to not leave the reader without any idea on how big the eavesdropping distances are, we give the following numbers, which as stated above are not valid in general at all, but can only serve to give a rough idea about these distances.

When a device is sending data in active mode, eavesdropping can be done up to a distance of about 10 m, whereas when the sending device is in passive mode, this distance is significantly reduced to about 1 m.

### 3.2 Data Corruption

Instead of just listening an attacker can also try to modify the data which is transmitted via the NFC interface. In the simplest case the attacker just wants to disturb the communication such that the receiver is not able to understand the data sent by the other device.

Data corruption can be achieved by transmitting valid frequencies of the data spectrum at a correct time. The correct time can be calculated if the attacker has a good understanding of the used modulation scheme and coding. This attack is not too complicated, but it does not allow the attacker to manipulate the actual data. It is basically a Denial of Service attack.

### 3.3 Data Modification

In data modification the attacker wants the receiving device to actually receive some valid, but manipulated data. This is very different from just data corruption.

The feasibility of this attack highly depends on the applied strength of the amplitude modulation. This is because the decoding of the signal is different for 100% and 10% modulation.

In 100% modulation the decoder basically checks the two half bits for RF signal on (no pause) or RF signal off (pause). In order to make the decoder understand a one as a zero or vice versa, the attacker must do two things. First, a pause in the modulation must be filled up with the carrier frequency. This is feasible. But, secondly, the attacker must generate a pause of the RF signal, which is received by the legitimate receiver. This means the attacker must send out some RF signal such that this signal perfectly overlaps with the original signal at the receiver's antenna to give a zero signal at the receiver. This is practically impossible. However, due to the modified Miller coding in the case of two subsequent ones, the attacker can change the second one into a zero, by filling the pause which encodes the second one. The decoder would then see no pause in the second bit and would decode this as a correct zero, because it is preceded by a one. In 100% modulation an attacker can therefore never change a bit of value 0 to a bit of value 1, but an attacker can change a bit of value 1 to a bit of value 0, in case this bit is preceded by a bit of value 1 (i.e. with a probability of 0.5).

In 10% modulation the decoder measures both signal levels (82% and Full) and compares them. In case they are in the correct range the signal is valid and gets decoded. An attacker could try to add a signal to the 82% signal, such that the 82% signal appears as the Full signal and the actual Full signal becomes the 82% signal. This way the decode would decode a valid bit of the opposite value of the bit sent by the correct sender. Whether the attack is feasible depends a lot on the dynamic input range of the receiver. It is very likely that the much higher signal level of the modified signal would exceed the possible input range, but for certain situations this cannot be ruled out completely.

The conclusion is that for the modified Miller encoding with 100% ASK this attack is feasible for certain bits and impossible for other bits, but for Manchester coding with 10% ASK this attack is feasible on all bits.

### 3.4 Data Insertion

This means that the attacker inserts messages into the data exchange between two devices. But this is only possible, in case the answering device needs a very long time to answer. The attacker could then send his data earlier than the valid receiver. The insertion will be successful, only, if the inserted data can be transmitted, before the original device starts with the answer. If both data streams overlap, the data will be corrupted.

### 3.5 Man-in-the-Middle-Attack

In the classical Man-in-the-Middle Attack, two parties which want to talk to each other, called Alice and Bob, are tricked into a three party conversation by an attacker Eve. This is shown in Figure 1.

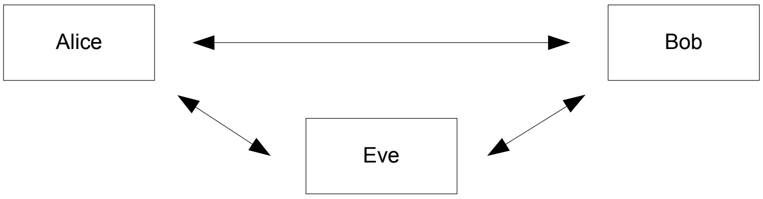

**Figure 1** Man-in-the-Middle Setup

Alice and Bob must not be aware of the fact that they are not talking to each other, but that they are both sending and receiving data from Eve. Such a setup is the classical threat in unauthenticated key agreement protocols like Diffie-Hellmann protocol. Alice and Bob want to agree on a secret key, which they then use for a secure channel. However, as Eve is in the middle, it is possible for Eve to establish a key with Alice and another key with Bob. When Alice and Bob later use their key to secure data, Eve is able to eavesdrop on the communication and also to manipulate data being transferred.

How would that work when the link between Alice and Bob is an NFC link?

Assuming that Alice uses active mode and Bob would be in passive mode, we have the following situation. Alice generates the RF field and sends data to Bob. In case Eve is close enough, she can eavesdrop the data sent by Alice. Additionally she must actively disturb the transmission of Alice to make sure that Bob doesn't receive the data. This is possible for Eve, but this can also be detected by Alice. In case Alice detects the disturbance, Alice can stop the key agreement protocol. Let's assume Alice does not check for active disturbance and so the protocol can continue. In the next step Eve needs to send data to Bob. That's already a problem, because the RF field generated by Alice is still there, so Eve has to generate a second RF field. This however, causes two RF fields to be active at the same time. It is practically impossible to perfectly align these two RF fields. Thus, it is practically impossible for Bob to understand data sent by Eve. Because of this and the possibility of Alice to detect the attack much earlier we conclude that in this setup a Man-in-the-Middle attack is practically impossible.

The only other possible setup is that Alice uses active mode and Bob uses active mode, too. In this case Alice sends some data to Bob. Eve can list and Eve again must disturb the transmission of Alice to make sure that Bob does not receive the data. At this point Alice could already detect the disturbance done by Eve and stop the protocol. Again, let us assume that Alice does not do this check and the protocol continues. In the next step Eve would need to send data to Bob. At first sight this looks better now, because of the active-active communication Alice has turned off the RF field. Now Eve turns on the RF field and can send the data. The problem here now is that also Alice is listening as she is expecting an answer from Bob. Instead she will receive the data sent by Eve and can again detect a problem in the protocol and stop the protocol. It is impossible in this setup for Eve to send data either to Alice or Bob and making sure that this data is not received by Bob or Alice, respectively.

We claim that due to the above given reasons it is practically infeasible to mount a Man-in-the-Middle attack in a real-word scenario.

# 4 Solutions and Recommendations

## 4.1 Eavesdropping

As described in section 3.1, NFC by itself cannot protect against eavesdropping. It is important to note that data transmitted in passive mode is significantly harder to be eavesdropped on, but just using the passive mode is probably not sufficient for most applications which transmit sensitive data.

The only real solution to eavesdropping is to establish a secure channel as outlined in section 4.6.

## 4.2 Data Corruption

NFC devices can counter this attack because they can check the RF field, while they are transmitting data. If an NFC devices does this, it will be able to detect the attack. The power which is needed to corrupt the data is significantly bigger, than the power which can be detected by the NFC device. Thus, every such attack should be detectable.

## 4.3 Data Modification

Protection against data modification can be achieved in various ways.

By using 106k Baud in active mode it gets impossible for an attacker to modify all the data transmitted via the RF link as described in section 3.3. This means that for both directions active mode would be needed to protect against data modification. While this is possible, this has the major drawback, that this mode is most vulnerable to eavesdropping. Also, the protection against modification is not perfect, as even at 106k Baud some bits can be modified. The two other options might therefore be preferred.

NFC devices can check the RF field while sending. This means the sending device could continuously check for such an attack and could stop the data transmission when an attack is detected.

The third and probably best solution would be a secure channel as described in section 4.6.

### 4.4 Data Insertion

There are three possible countermeasures. One is that the answering device answers with no delay. In this case the attacker cannot be faster than the correct device. The attacker can be as fast as the correct device, but if two devices answer at the same time no correct data is received.

The second possible countermeasure is listening by the answering device to the channel during the time, it is open and the staring point of the transmission. The device could then detect an attacker, who wants to insert data.

The third option again is a secure channel between the two devices.

### 4.5 Man-in-the-Middle-Attack

As already outlined in section 3.5 it is practically impossible to do a Man-in-the-Middle-Attack on an NFC link. The recommendation is to use active-passive communication mode such that the RF field is continuously generated by one of the valid parties. Additionally, the active party should listen to the RF filed while sending data to be able to detect any disturbances caused by a potential attacker.

### 4.6 Secure Channel for NFC

Establishing a secure channel between two NFC devices is clearly the best approach to protect against eavesdropping and any kind of data modification attack.

Due to the inherent protection of NFC against Man-in-the-Middle-Attacks it is rather easy and straightforward to setup a secure channel.

A standard key agreement protocol like Diffie-Hellmann based on RSA [4] or Elliptic Curves could be applied to establish a shared secret between two devices. Because Man-in-the-Middle is no threat, the standard, unauthenticated version of Diffie-Hellman works perfectly.

The shared secret can then be used to derive a symmetric key like 3DES or AES, which is then used for the secure channel providing confidentiality, integrity, and authenticity of the transmitted data. Various modes of operation for 3DES and AES could be used for such a secure channel and can be found in literature [3].

#### 4.6.1 NFC Specific Key Agreement

Besides the standard key agreement mechanism, it is also possible to implement an NFC specific key agreement. This one does not require any asymmetric cryptography and therefore reduces the computational requirements significantly. Theoretically, it also provides perfect security.

The scheme works with 100% ASK only and it is not part of the ISO standard on NFC. The idea is that both devices, say Device A and Device B, send random data at the same time. In a setup phase the two devices synchronize on the exact timing of the bits and also on the amplitudes and phases of the RF signal. This is possible as devices can send and receive at the same time. After that synchronisation, A and B are able to send at exactly the same time with exactly the same amplitudes and phases.

While sending random bits of 0 or 1, each device also listens to the RF field. When both devices send a zero, the sum signal is zero and an attacker, who is listening, would know that both devices sent a zero. This does not help. The same thing happens when both, A and B, send a one. The sum is the double RF signal and an attacker knows that both devices sent a one. It gets interesting once A sends a zero and B sends a one or vice versa. In this case both devices know what the other device has sent, because the devices know what they themselves have sent. However, an attacker only sees the sum RF signal and he cannot figure out which device sent the zero and which device sent the one. This idea is illustrated in Figure 2. The top graph shows the signals produced by A in red and by B in blue. A sends the four bits: 0, 0, 1, and 1. B sends the four bits: 0, 1, 0, and 1. The lower graph shows the sum signal as seen by an attacker. It shows that for the bit combinations (A sends 0, B sends 1) and (A sends 1, B sends 0) the result for the attacker is absolutely the same and the attacker cannot distinguish these two cases.

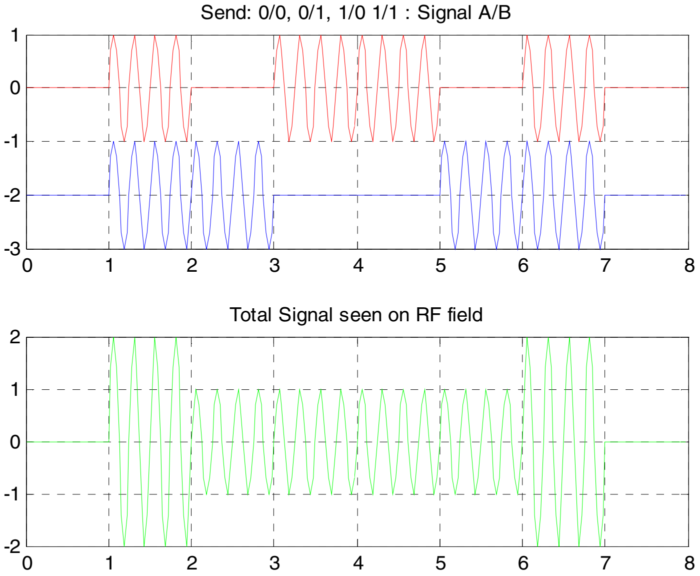

**Figure 2** NFC specific Key Agreement

The two devices now discard all bits, where both devices sent the same value and collect all bits, where the two devices sent different values. They can either collect the bits sent by A or by B. This must be agreed on start-up, but it doesn't matter. This way A and B can agree on an arbitrary long shared secret. A new bit is generated with a probability of 50%. Thus, the generation of a 128 bit shared secret would need approximately 256 bits to be transferred. At a baud rate of 106 kBaud this takes about 2.4 ms, and is therefore fast enough for all applications.

The security of this protocol in practice depends on the quality of the synchronisation which is achieved between the two devices. Obviously, if an eavesdropper can distinguish data sent by A from data sent by B, the protocol is broken. The data must match in amplitude and in phase. Once the differences between A and B are significantly below the noise level received by the eavesdropper the protocol is secure. The level of security therefore also depends on the signal quality at the receiver. The signal quality however again depends on many parameters (e.g. distance) of the eavesdropper. In practice the two devices A and B must aim at perfect synchronisation. This can only be achieved if at least one of A or B is an active device to perform this synchronization.

Note, that in a recently published paper [5], the same idea for key agreement between an RF reader and an RF tag is presented in a slightly different setup. The paper uses a special so-called noisy tag. This noisy tag is a standard RFID tag, which acts as a third party inserting random looking bits into the communication from the real tag to the real reader. The reader however can calculate the bits sent by the noisy tag and can then calculate the bits sent by the real tag. The problem we see with this approach is that the noisy tag will not be able to do any synchronization with the real tag. This would be too complicated for a simple tag. Therefore, we think that this approach cannot work in practice. It would require a more sophisticated noisy device instead of the noisy tag to run that protocol in a secure way.

## 5 Conclusion

We presented typical use cases for NFC interfaces. A list of threats has been derived and addressed. NFC by itself cannot provide protection against eavesdropping or data modifications. The only solution to achieve this is the establishment of a secure channel over NFC. This can be done very easily, because the NFC link is not susceptible to the Man-in-the-Middle attack. Therefore, well known and easy to apply key agreement techniques without authentication can be used to provide a standard secure channel. This resistance against Man-in-the-Middle attacks makes NFC an ideal method for secure pairing of devices. Additionally, we introduced an NFC specific key agreement mechanism, which provides cheap and fast secure key agreement.

## References

[1] "Information technology - Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1)", ISO/IEC 18092, First Edition, 2004-04-01.

[2] Klaus Finkenzeller, "RFID Handbuch", Hanser Verlag, 2002.

[3] Morris Dworkin, "Recommendation for Block Cipher Modes of Operation", NIST Special Publication 800-38A, 2001.

[4] W. Diffie and M.E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory 22 (1976), 644-654.

[5] C. Castelluccia and G. Avoine, "Noisy Tags: A Pretty Good Key Exchange Protocol for RFID Tags", Proceedings of CARDIS 2006, LNCS 3928, 289-299, 2006.