PREVENTING MAN-IN-THE-MIDDLE ATTACKS IN NEAR FIELD COMMUNICATION BY OUT-OF-BAND KEY EXCHANGE

A Thesis

by

SRI ADITYA PANDA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Riccardo Bettati
Co-Chair of Committee,	A. L. Narasimha Reddy
Committee Member,	Paul Gratz
Head of Department,	Miroslav M. Begovic

May 2016

Major Subject: Computer Engineering

ABSTRACT

Near Field Communication (NFC) is an RFID based proximity communication technology. The extensive use of NFC technology for popular and sensitive applications such as financial transactions and content sharing necessitates the implementation of secure transmission standards for data exchange. NFC-SEC is one such set of cryptographic standards that extends NFC to provide better security. However, NFC is still susceptible to Man-in-the-Middle (MITM) attacks due to the lack of device authentication, which in turn allows for masquerading and other attacks. Inclusion of a certification authority has commonly been proposed to resolve this issue at the cost of significant additional communication overhead. In this thesis, we first demonstrate a practical MITM attack on an NFC-SEC communication session. We then present NonceCrypt, a light-weight countermeasure against this class of attacks. NonceCrypt addresses the vulnerability of NFC-SEC by an added step of authentication over a secure out-of-band communication channel. We implement NonceCrypt on an Arduino platform and evaluate its implementation cost and runtime overhead in a set of experiments. Results indicate that the increase memory and time overhead for this scheme are negligible. It avoids involving any additional entities in the communication and is based on a flexible implementation scheme that can be used for both smartphones and contactless cards.

DEDICATION

To my parents.

ACKNOWLEDGEMENTS

I am grateful to my advisor, Dr. Riccardo Bettati, for his guidance and support throughout the project. I thank him for spending long hours to meet with me and patiently mentoring me through the challenging phase of the project. His faith in me was immensely reassuring.

I am indebted to my parents for encouraging me to pursue my goals. I would like to thank my family for standing by me and providing me with all the resources that I needed through the course of my study. My friends boosted my morale, motivating me to stay the course and persevere.

I am thankful to the other members of my committee, Dr. Narasimha Reddy and Dr. Paul Gratz for their time and invaluable insights to enhance the project.

NOMENCLATURE

AES	Advanced Encryption Standard
CM	Centimeters
ECDH	Elliptic Curve Diffie-Hellman
ISO	International Organization for Standardization
KB	Kilo Bytes
MAC	Message Authentication Code
MHz	Megahertz
MITM	Man-in-the-Middle
MS	Milliseconds
NFC	Near Field Communication
RFID	Radio-Frequency Identification
SRAM	Static Random Access Memory

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Near field communication (NFC) [24] is a widely used proximity communication technology. It is an RFID based radio communication [24] with a frequency of 13.56MHz and a typical operating distance of 10cm or less. It is most commonly used to exchange a small amount of data or for authentication to communicate using another technology such as Bluetooth [22]. Its applications include contactless payment systems such as Android Pay [3], Apple Pay [4], Google Wallet [18] and NFC enabled credit card systems [23], e-passports [25], social networking for content sharing [6] and as smart card access tokens [26].

All NFC communication requires the participation of two entities, an initiator and a target. The *initiator*, also known as the *reader*, initiates the communication by selecting a target and communicating with it. The *target*, also known as the *tag*, responds with its data when selected. NFC communication may occur in two forms: passive or active. In *passive* communication, the initiator can generate its own power but the target cannot. The target is powered by the RF field of the initiator, i.e., when the target is within the RF range of the initiator, then the former uses the power from the RF field to communicate. The target loses power and returns to idle state when it is out of range of the initiator. In *active* communication, both the initiator and the target generate their own power and communicate by intermittently turning the RF field on to send data and off to receive data.

With the increase in the use of NFC for sensitive applications such as monetary transactions [3, 4, 18, 23], health applications [28, 29, 35] and electronic passports [25], it is important to shield the communication against any possible security threat. NFC security is based on the premise that, as the devices need to be in close prox-

imity, it is not practical to carry out a number of attacks such as eavesdropping or man-in-the-middle attacks [22]. Additional standards have been proposed to establish secure communication using NFC [7, 8]. In the following sections, we explore the various threats possible on NFC and the safeguards provided by these standards.

1.1 Motivation

Multiple authors have demonstrated that the eavesdropping range of NFC communication can be extended to one meter and beyond [20, 22]. The increase in the range over which NFC communication can be observed, in turn increases the chances of attacks that had been previously ruled out due to lack of proximity. Attacks such as eavesdropping on the communication, relay attacks and denial of service have been proven to be successful [13, 20, 21, 31]. These attacks have been carried out with minimal additional equipment. Other attacks, such as data modification, are much more complicated, but nevertheless possible to some extent [22]. These attacks successfully manipulate the data by changing the bits from 0 to 1 and vice versa under some coding schemes.

Eavesdropping attacks are the simplest to carry out and only require an antenna, or an additional reader, to receive the communication. This could lead to invasion of privacy and leakage of sensitive information if the information transmitted via NFC is in plain text. *Relay* attacks require a proxy initiator to communicate with the genuine target and relay the information to a proxy target, which in turn communicates with a genuine initiator. Attackers can use relays to connect initiators and targets that are not in close proximity, thus invalidating a basic premise of the security paradigm for NFC. For example, this allows to establish connections and process transactions between a reader and the NFC enabled credit card of an unsuspecting owner. Strict timing constraints have been proposed as a solution to protect against relay attacks.

This solution has issues in implementation as different devices have different timing requirements, and it is hard to impose a universal timing constraint [14]. Denial-of-Service attacks can be implemented by either corrupting the data by introducing an additional RF field that just transmits random information at 13.56 MHz or by keeping the initiator or target busy to prevent them from establishing communication with one another [22, 31].

A suite of standards, collectively known as NFC-SEC, have been proposed to reduce the effectiveness of attacks by establishing a secure channel [7, 8, 9, 10, 11]. The standards NFC-SEC-01 [8] and NFC-SEC-02 [9] describe the establishment of a secure channel by a key exchange based on the Elliptic Curve Diffie-Hellman (ECDH) protocol, followed by data encryption using the Advanced Encryption Scheme (AES). NFC-SEC-03 [10] and NFC-SEC-04 [11] provide key agreement using asymmetric and symmetric cryptography respectively. Specifically, NFC-SEC-03 [10] requires certification provided by a trusted third party and NFC-SEC-04 [11] requires the establishment of a pre-shared key, which may not be feasible. NFC-SEC-02 [9] is a variation of NFC-SEC-01 [8]. Given the limited practicality of NFC-SEC-03 and NFC-SEC-04, we limit the discussion to NFC-SEC-01 [8].

In NFC-SEC-01 [8], on establishment of a secure channel by Elliptic Curve Diffie-Hellman, all the information that is transmitted by the two entities is encrypted using a shared secret key, which is known only to the entities. This rules out the possibility of eavesdropping attacks, as the attacker would not be able to read the encrypted message [12]. In case of a relay attack, the relaying entity would not be able to compute the secret key and decrypt the transmission to obtain the original message. Here, message integrity may be kept intact, but there is no provision to verify if the communicating device is authentic or a malicious entity relaying the communication from an authentic device. Hence, NFC-SEC-01 alone cannot prevent relay attacks.

While encryption would ensure message authentication, and so prevent the attacker from modifying message payloads, additional mechanisms are required in order to prevent relay channels to be established.

1.2 Man-in-the-Middle Attacks in NFC

The NFC-SEC-01 protocol restricts many attacks that are otherwise possible with no encryption mechanism. The proposed key exchange mechanism for NFC-SEC-01 is based on a basic Diffie-Hellman scheme. Since this scheme does not consider authentication of the two participants, this leaves it susceptible to man-in-the-middle (MITM) attacks [12]. It has been argued that the proximity requirements for NFC rule out the possibility for MITM attacks. Unfortunately, there exist scenarios where the need for physical proximity alone does not secure the communication. In a normal NFC communication scenario (Figure 1.1), a target (e.g. an NFC capable credit card) triggers the communication when it comes in close proximity of an initiator (a credit card reader). Immediate proximity of the target to the initiator in implicitly assumed by the credit card reader, and the communication is deemed as secure. One can envision a scenario the genuine initiator is blocked from directly communicating with the genuine target despite the latter's proximity. This in turn would lead to a possibility for an MITM attack. For example, an attacker can block the initiator by methods as simple as covering it with a Faraday cage or by forcing it to remain busy using a denial of service attack [21]. Figure 1.2 depicts such a scenario.

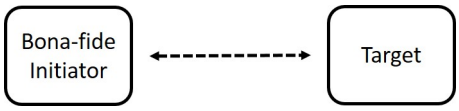

Figure 1.1: Normal NFC Communication Scenario.

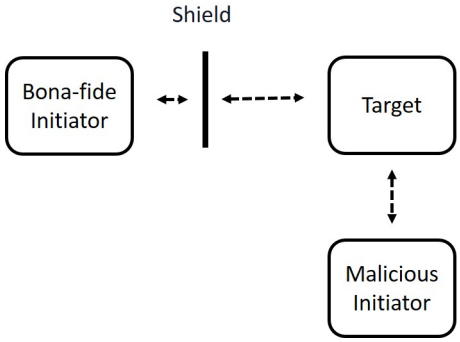

Figure 1.2: MITM Attack Scenario by Blocking Communication Between the Genuine Initiator and Target Using a Shield, Such as a Faraday Shield.

The MITM attack can be carried out by establishing two separate sessions with a genuine initiator and a genuine target, as shown in Figure 1.3. As both the sessions have been established with the attacker, the attacker has the secret key for both the sessions. She can therefore use the secret key of the session with the initiator to decrypt the message. The original message, which is now in plain text format, may now be modified and encrypted using the other secret key. The message is then passed on to the genuine target. As the protocol does not provide for any form of signatures or certification authority, none of the entities involved in the communication can determine the authenticity of the other entities. This creates a situation where the entities may establish a secure channel with a deceptive entity, which voids the effectiveness of the protocol.

We note that several variations of the scenario depicted in Figure 1.3 can be constructed. For example, a malicious target may contact the bona-fide initiator. In this case, the attacker can eavesdrop on the encrypted conversation and/or modify the content of the communication. This has been depicted in Figure 1.4. Alternatively, a malicious initiator can directly act as a masquerading agent, without the need for an additional bona-fide initiator, and thereby hijack the communication with the bona-

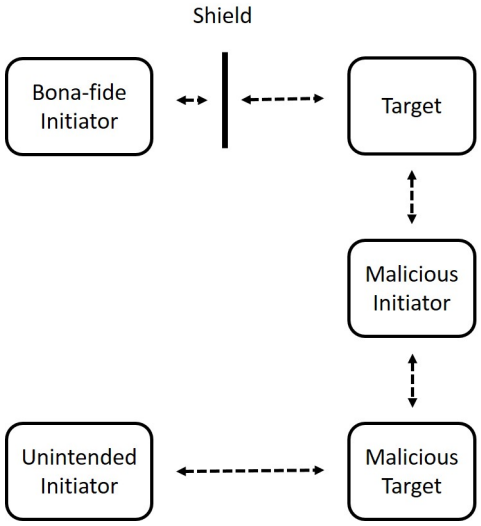

Figure 1.3: MITM Attack Scenario with an Unintended Initiator.

fide target as depicted in Figure 1.2. This is possible because the bona-fide target does not realize that the bona-fide initiator has been blocked from communicating with it by using a shield (e.g. Faraday cage), and presumes that it is communicating with the bona-fide initiator instead of realizing that it is actually communicating with a malicious initiator. This leads to a masquerading attack where the malicious initiator pretends to be the bona-fide initiator to communicate with the target.

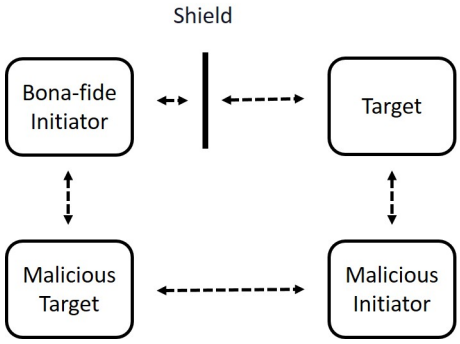

Figure 1.4: MITM Attack Scenario with the Original Initiator.

In summary, the lack of authentication of the communicating NFC devices opens the door to a number of MITM attacks, ranging from eavesdropping to masquerading.

2. SOLUTION CRITERIA

Clearly, a solution is needed to reliably authenticate the participants in a secure NFC communication. As we pointed out in Chapter 1, a number of approaches exist, such as NFC-SEC-03 and NFC-SEC-04, which have limited value in practice. The reliance of NFC-SEC-03 on certification authorities is a major impediment in its applicability to services that cannot contact the authority securely or without additional hardware. Similarly, NFC-SEC-04 uses a pre-shared key for authentication, which leaves it susceptible to side-channel attacks. This chapter reflects on typical usage scenarios for NFC to identify a set of criteria that any proposed solution needs to satisfy in order to be viable.

2.1 Security

Any proposed solution needs to implement better security than NFC-SEC-01 [8] to prevent MITM attacks while not compromising on other security features, such as the encryption of data using a shared secret. NFC-SEC-01 [8] provides security against attacks such as eavesdropping and data modification. The proposed solution must not compromise the security provided against such attacks. Any untrustworthy communication channels must be avoided.

2.2 Deployability

The solution must be easily deployable and must be cost-effective. Solutions that require additional equipment be attached to the device or any additional hardware for its implementation may not be very adaptable as they may cause inconvenience to the user. Also, additional hardware for implementation would mean that there would be extra costs for the solution to be implemented. Even then, the solution

may not be compatible with the present device models and can be implemented only in newer versions of the devices, which may cause a drastic reduction in the user base.

2.3 Compatibility

The proposed solution must be compatible with the present usage model. It must be viable for use in the different applications that already use NFC and must have scope to be extended to future applications that may implement NFC transactions as well. This implies that the solution must work both on active devices, such as phones, as well as on passive devices, such as smart cards. The implementation must not be computation intensive or require additional hardware, as this would hinder its implementation on passive device platforms.

2.4 Overhead

NFC communications typically have transaction times in the order of milliseconds. The proposed solution must aim at keeping the time overhead as low as possible so that they do not inconvenience the user. Also, most transactions require that the interacting devices be kept in very close proximity, so increasing the transaction time may be cumbersome. Solutions that suggest interaction with an additional entity, such as a third-party trusted source, would add considerably to the overhead.

2.5 Ease of Use

Ease of use is an important criterion that decides the success of an application. Solutions that require the user to interact in the transaction process, in addition to holding the devices in proximity to facilitate NFC communication, may be cumbersome. Examples of such interactions may be requiring the user to enter a pass code or even a considerable increase in the time for which the devices must be held

in proximity. We note that it is also important that security of the application is given priority over concerns of minor unease. There is a need to maintain a balanced trade-off between security and user-friendliness.

The defined solution criteria provides a framework to build an efficient solution. While solutions have been presented by various authors to address similar issues, there is still a necessity for a comprehensive solution that addresses all the aforementioned criteria. In Chapter 3, we identify drawbacks in the solutions offered by different authors and present a comprehensive solution in Chapter 4. This solution is later evaluated against the criteria to analyze its efficiency.

3. RELATED WORK

Various solutions have been proposed to secure NFC communication by the inclusion of a certification authority. The authors have suggested the addition of a Certification Authority in [2]. They have suggested that adding authentication to the Diffie-Hellman protocol can prevent MITM attacks. This is done by communicating with a server to generate a signed message. Although this scheme provides better security over a conventional Diffie-Hellman protocol, the added security comes at the cost of added transaction time for communication with the server. Also, this would require the system to be online and there must be uninterrupted access to the server. The scheme also requires a lot of added infrastructure in the form of an online server to manage the certification requests. The inclusion of a digital certificate embedded in an inaccessible part of the tag for offline tag authentication has been suggested in [33]. This proposed scheme is a solution that is specific to attacks that use cloned tags and is not extended to other scenarios. In [27], the authors have suggested registration with an authentication center to avoid MITM attacks. In many situations, the inclusion of a certification authority is not an effective solution, as this would require the establishment of a server that can constantly process the multiple requests for public keys and verification of identity. This would greatly increase the communication overhead and the time for the transaction. Also, it would have to rely on another communication channel such as internet or SMS, which may be unreliable. The use of a secure Bluetooth based application for credit transfer has been suggested in [30]. This approach assumes that the Bluetooth channel is secure. It also requires that a network operator be contacted to evaluate the validity of the transaction. The approach does not specify and cryptographic measures to secure

the communication in any of the channels.

In [12], the authors suggest the use of pseudonyms instead of public keys to protect the privacy of users. This is done by additional cryptographic processing to generate the pseudonym. Unfortunately, this comes at an additional cost of storage space, computation time and transference time. This, in turn, leads to an increase in the transaction time. In addition, while the privacy of the user is protected, there is no additional security against MITM attacks. A system called Dhwani has been proposed in [32], which is a "secure peer-to-peer acoustic NFC" communication technique that uses JamSecure. The idea is to jam the signal and then use self-interference cancellation to recover the signal at the receiver. This approach requires the use of a microphone and receiver. This precludes the use of this technique in passive tags. Also, shielding the receiver, directional reception or suitable placement of a malicious receiver may compromise the data.

The authors in [19] have proposed an improved NFC-SEC scheme for NFC-based health care applications. The proposal involves the establishment of public key infrastructure including a certification authority for providing signed communication exchanges. This is to eliminate any possible errors as health care is a critical application. This proposal requires establishment of expensive additional infrastructure which may not be necessary for other applications. Another health care application based on NFC has been suggested in [34], which also suggests the use of a server for verifying authenticity. But, all the communications do not need to contact the server to verify authenticity and there is limited security on such channels, which leaves a good scope for attacks.

4. SOLUTION

A major drawback of the NFC-SEC-01 protocol is that it does not offer authentication, but rather relies on physical proximity to ensure authentication. We showed in Chapter 1, a number of scenarios where the security of NFC-SEC communications are violated despite physical proximity. The proposed solution rectifies this issue by providing an ad-hoc off-line authentication scheme. The proposed solution shall henceforth be referred to as "NonceCrypt".

NonceCrypt takes advantage of the typical usage scenario for NFC, where a user tries to make a payment at an NFC terminal. The user would then be asked to enter a nonce, randomly generated by the user's device, into the NFC terminal before the transaction can proceed. Now, both the participating entities shall have this nonce. This nonce, henceforth known as "ChannelKey", can be used as a shared key exchanged over a secure channel. To ensure that the ChannelKey is exchanged securely, an out-of-bound channel may be used for its exchange. We will further discuss the exchange of ChannelKey later in this chapter. This ChannelKey can be used to encrypt and decrypt all the communication that is part of the NFC-SEC-01 protocol. As the ChannelKey is only shared by the authentic initiator and target, and is a random nonce generated for each transaction, it is not possible for any entity which does not already have the ChannelKey to establish the NFC-SEC protocol with ether of the authentic entities.

4.1 Exchange of ChannelKey

The ChannelKey must be exchanged over a secure channel, for which we suggest the use of an out-of-band communication channel. This is because, if the ChannelKey were to be exchanged over the same NFC channel, then a malicious entity would

be able to eavesdrop over the communication. Once the malicious entity has the ChannelKey, it could successfully attack the communication as before. This makes the security of the exchange channel a critical component. Other wireless channels such as Bluetooth may be used for this operation, but their vulnerability makes them unsuitable for this transaction. The channels suggested for use would be the visual band (for example, a QR code generated on the tag is read by a camera on the reader) or a tactile channel where the user has to enter a value by hand. These channels are deemed as more secure as they involve direct user interaction and could not be imitated by a device as easily.

4.2 Encryption with ChannelKey

The ChannelKey could be used to provide an added layer of encryption. This is done by using the ChannelKey to derive an encryption key or by using it directly as an encryption key. As a malicious entity would not have access to the ChannelKey, the data cannot be correctly decrypted by the malicious entity. For instance, the public key of an authentic entity is encrypted by the ChannelKey before being sent to the malicious entity. Similarly, the public key of the malicious entity is expected to be encrypted by the ChannelKey, which is not possible. So, during the computation of the shared secret, the decrypted public keys would be different from the original public keys, which would result in different shared secrets for both entities. This would ensure that none of the communication received by the malicious entity is authentic and the communication would break down at the key confirmation phase of NFC-SEC-01.

The ChannelKey is generated on each attempt and should be cryptographically unrelated to any previous ChannelKey. This would ensure that there is no chance for a replay attack.

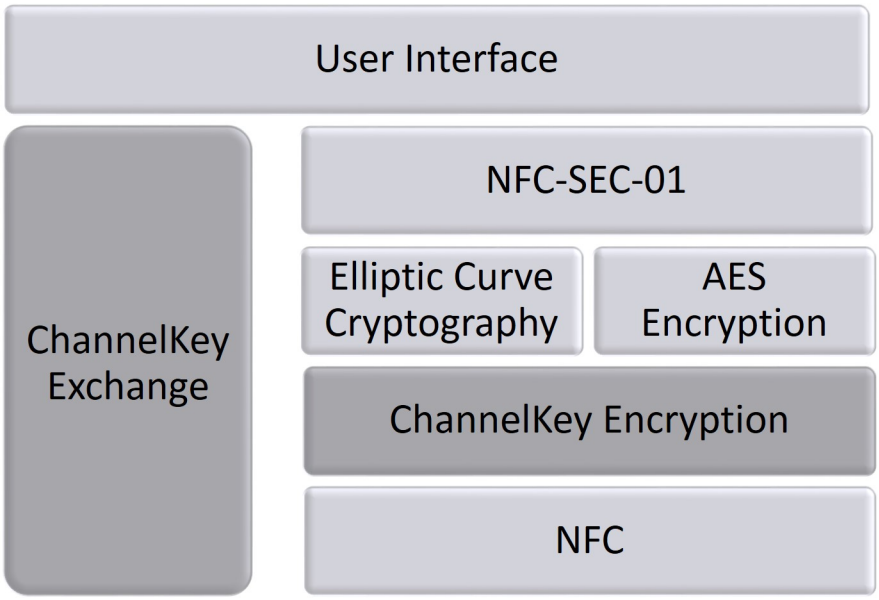

Figure 4.1: NonceCrypt Architecture.

4.3 NonceCrypt Architecture

The architecture of NonceCrypt is described in Figure 4.1. The NonceCrypt system has two more layers in addition to those present in NFC-SEC-01. When a message is received from the user to initiate NFC communication, it first performs a ChannelKey exchange through a secure channel. Once the ChannelKey has been exchanged, the NFC communication is initiated. The NFC-SEC-01 layer initially works with the Elliptic Curve Cryptography layer in establishing a shared secret using the Elliptic Curve Diffie-Hellman protocol. The NFC-SEC-01 layer also interacts with the AES encryption layer to generate multiple derived keys for encryption and decryption, and to verify the integrity of the communication by checking the MAC. All outgoing messages are encrypted using AES encryption before being transferred over NFC. Similarly, incoming messages are decrypted before being passed on to the user. The ChannelKey Encryption layer encrypts outgoing communication and

decrypts incoming communication with the ChannelKey. The NFC layer forms the lowest layer of the communication and is used to send and receive messages over the NFC channel.

5. IMPLEMENTATION

In order to demonstrate the efficacy of the NonceCrypt approach, we implemented and evaluated it on an Arduino platform. We first implemented a basic NFC-SEC-01 system. We then performed an MITM attack to demonstrate the vulnerability of the NFC-SEC-01 approach. Once the vulnerability of NFC-SEC-01 had been clearly demonstrated, we implemented NonceCrypt as an additional layer over the present NFC-SEC-01 system.

5.1 Equipment

We selected the Arduino Uno [5] platform as a demonstration vehicle because of its broad availability, the availability of an easy-to-program NFC infrastructure and its ability to perform the relatively heavy computation for NFC-SEC-01 and the NonceCrypt extension. The Arduino Uno [5] is an ATmega328P based 8-bit AVR microcontroller board. It has 32KB of programmable Flash memory and a 2KB SRAM. It is connected to a PN532 NFC transceiver [1] to be able to communicate using NFC. The PN532 NFC breakout board [1] operates at 13.56 MHz and supports both reader/writer mode and card emulation for ISO14443A.

The software for the system needed to have a very low memory footprint in order to be used with the hardware. Hence, we chose the libraries with this parameter in perspective. Also, we modified all the libraries to optimize their performance and include the necessary features. We modified the Adafruit-PN532 library [15] substantially to implement NFC functions and communicate with the PN532 boards. This library provided the basic NFC functionality. Tag emulation had to be added to the library to implement an NFC-SEC-01 card. We chose micro-ecc library [16] to implement Elliptic Curve Cryptography needed to perform key exchange as it had a

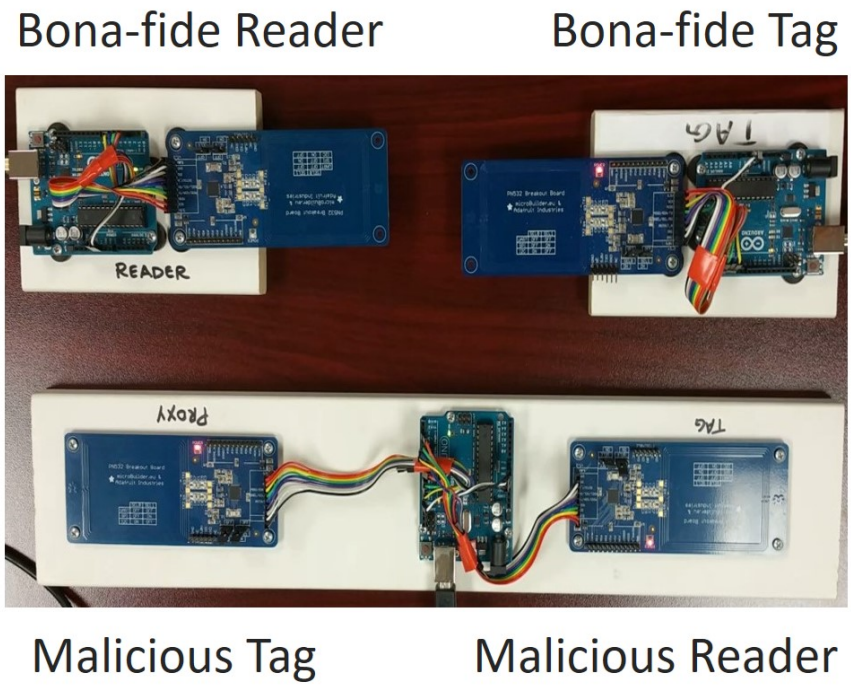

Figure 5.1: Equipment Used for Implementation of the Reader, Tag and Malicious Entity.

very low memory footprint. We used tiny-AES-128-C library [17] to implement AES encryption for key derivation and data encryption. We added AES-XCBC-PRF-128, AES-XCBC-MAC-96 and AES-CTR modes to the library to perform key derivation, MAC derivation and encryption respectively. Figure 5.1 shows the equipment used for the implementation.

5.2 Implementation of NFC-SEC-01 System

We implement an NFC reader and a tag, each created by connecting an Arduino Uno to a PN532 board. Both the entities have the NFC-SEC-01 protocol built into them. The base layer consists of the NFC communication layer, which defines the functionality of the reader and the tag. Key exchange, derivation, and confirmation

are built into the data exchanges between the reader and tag as part of the NFC-SEC-01 layer. Performing a MITM attack on this system requires an additional entity, which can be modeled as a malicious reader connected to a malicious tag. The malicious reader and tag function similar to the genuine entities, except that they can exchange and modify data. This is accomplished by having two PN532 boards, one behaving as a reader and the other as a tag, connected to a single Arduino Uno platform that provides a medium to exchange data, as shown in Figure 5.1. During an MITM attack, communication is channeled from a genuine tag to a malicious reader, which receives the plain text data, modifies it and transmits it from malicious tag to the genuine reader.

5.3 Implementation of NonceCrypt System

The NonceCrypt system can be implemented using a few additions to the NFC-SEC-01 system. A few additional layers need to be included in the software while keeping the hardware implementation constant for the reader, tag and the malicious entity. As part of the NonceCrypt system, ChannelKey needs to be exchanged prior the communication. A layer of encryption then needs to be added between the NFC-SEC-01 and NFC communication layers such that any message generated in the NFC-SEC-01 layer is encrypted before sending over the NFC communication layer.

For simplicity, a hard-coded ChannelKey is used on the reader and the tag, and a slightly different ChannelKey is used for the malicious entity to mimic the secure transfer of the ChannelKey from the reader to the tag without the knowledge of the malicious entity. Also, all encryptions can be performed using a simple XOR cipher. The implementation of the NFC-SEC-01 system using AES encryption can have a huge memory footprint, that may be reduced by the use of XOR encryption, as presented in Table 5.1. The reader requires 13kB of program storage space and 1.4kB

of dynamic memory when using AES encryption, which can be reduced to 10kB and 562 bytes respectively, by using XOR encryption. The total program storage space is 32kB and dynamic memory is 2kB.

Entity	AES Encryption		XOR Encryption	
	Flash memory	SRAM	Flash memory	SRAM
Reader	13590 (42%)	1400 (68%)	10430 (32%)	562 (27%)
Tag	13568 (42%)	1349 (65%)	10824 (33%)	601 (29%)

Table 5.1: Comparison of memory footprint for NFC-SEC-01 implementations with AES encryption and XOR encryption.

In Table 5.2, the memory footprint of the NFC-SEC-01 system is compared to the memory footprint of the NonceCrypt system. The increase in memory usage is almost negligible with an increase of 140 bytes for program storage space and 24 bytes for dynamic memory for the reader. The increase in memory for the tag is 90 bytes of program storage space and 16 bytes of dynamic memory.

Entity	NFC-SEC-01 System		NonceCrypt System	
	Flash memory	SRAM	Flash memory	SRAM
Reader	10430 (32%)	562 (27%)	10570 (32%)	586 (28%)
Tag	10824 (33%)	601 (29%)	10914 (33%)	617 (30%)
Malicious	12110 (37%)	689 (33%)	12310 (38%)	705 (34%)

Table 5.2: Comparison of memory footprint for NFC-SEC-01 system and NonceCrypt system implementations with XOR encryption.

6. EVALUATION

In this chapter, we evaluate the strengths and weaknesses of NFC-SEC-01 and NonceCrypt against the Criteria provided in Chapter 2. We analyze NFC-SEC-01 first and later evaluate how NonceCrypt improves on the flaws present in the former. We also discuss how the flexibility of NonceCrypt is a great advantage that lets it be easily adapted to suit the application.

6.1 Evaluation of NFC-SEC-01

6.1.1 Security

NFC-SEC-01 provides security by encrypting data with a key available only to the participating entities. Hence, any communication between the entities is encrypted and cannot be availed by any other entity, such as an eavesdropper. This provides security against eavesdropping. Also, in case of a relay attack, the communication may be relayed without the relay channel gaining any knowledge of the communication. But, this form of security relies on the assumption that the participating entities are authentic. There is no entity authentication to verify that the participating entities are indeed authentic. If one of the participating entities is malicious, the encryption can be broken and plain text data may be obtained. This data may be eavesdropped upon or manipulated. This is a major drawback in the scheme. NonceCrypt resolves this by providing entity authentication.

6.1.2 Deployability

NFC-SEC-01 can be implemented in software and integrated into the present NFC system as specifies in [7]. Every device has a choice of whether to include or exclude NFC-SEC-01 as part of the protocol. It can be implemented on both active

and passive devices. NFC-SEC-01 would require some additional computation power as compared to a simple NFC transaction because of the key exchange and encryption components involved.

6.1.3 Compatibility

NFC-SEC-01 is implemented as part of the NFC transaction and can occur without any interaction from the user. Hence, it would be compatible with any current user model for NFC, which involves a tag to be placed against a reader for a transaction to occur.

6.1.4 Overhead

The time overhead for NFC-SEC-01 increases in comparison to a normal NFC transaction. This is because of additional computation time required for key derivation and encryption. There is also an increase in the number of exchanges between the sender and receiver for the establishment of a secure channel, which causes an increase in the overhead time. The overhead time can be approximated as 6.3 seconds.

6.1.5 Ease of Use

NFC-SEC-01 is very user friendly as there is no required user interaction. Establishment of the secure channel is carried out as part of the NFC transaction and may occur without the knowledge of the user.

6.2 Evaluation of NonceCrypt

6.2.1 Security

NonceCrypt provides better security than simply using NFC-SEC-01 by the inclusion of a separate authentication factor. NFC-SEC has inherent protection against eavesdropping, and its major vulnerability, which is MITM attacks. NonceCrypt

provides an authentication mechanism by which the only the genuine entities can decrypt the encrypted public keys. As the ChannelKey is a random value that changes for every transaction, it is not easy to guess it based on the encrypted data. It differs in this aspect from a constant password. A critical component of NonceCrypt is the ability to transfer the ChannelKey over a secure channel. Though this could be done by methods such as transmission over another wireless channel, capturing photographs via cameras, bar codes, the most secure method would be to include user interaction as, in the other methods, there could be a chance for manipulation using advanced equipment. This implementation detail can be decided based on the application. The ChannelKey is never transmitted as plain text through an NFC channel and is only used to encrypt the communication thereby nullifying the risk of a malicious entity eavesdropping to recover it. Also, as all the data is encrypted using the ChannelKey, it would protect the genuine entity from establishing a secure channel with the malicious entity by terminating the communication at the key confirmation phase.

6.2.2 Deployability

NonceCrypt is an entirely software implementation except for the method of exchange of the ChannelKey. So, any NFC system that is already present can simply be updated to improve its security. The method of exchange of ChannelKey would be decided based on the application. Most ideas are easy to implement on active NFC platforms such as smart phones and issue would arise out of implementation constraints on passive devices such as smart cards. But, as the system does not require much more computation power or storage space than NFC-SEC-01 already does, it would not present much of a challenge.

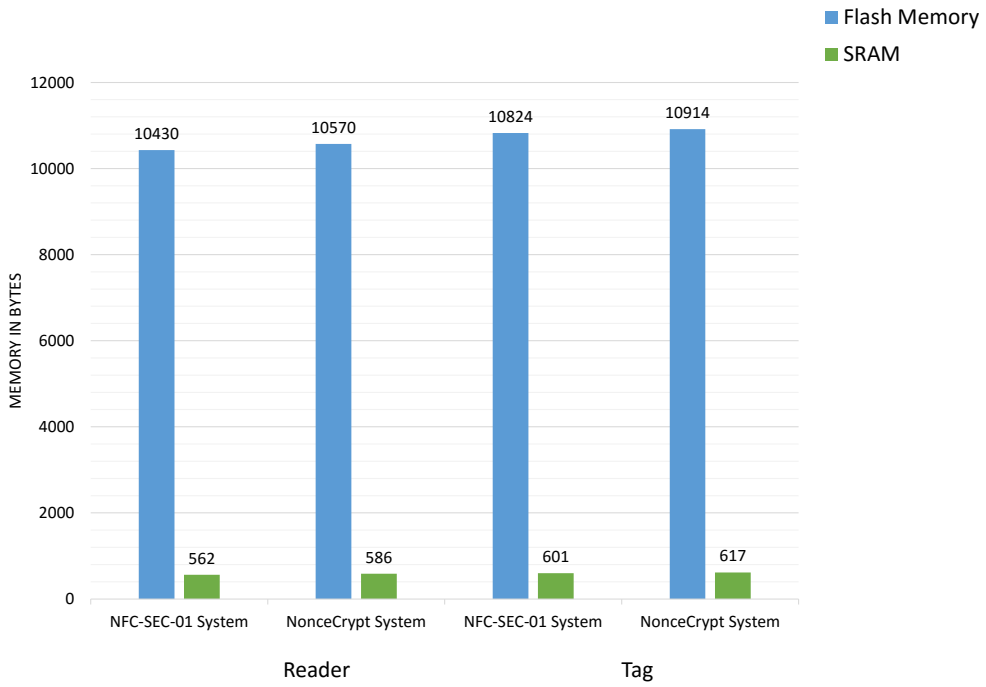

Figure 6.1: Comparison of Memory Requirements for the Reader and Tag of the NFC-SEC-01 System and NonceCrypt System.

6.2.3 Compatibility

The flexibility of implementation provided by NonceCrypt is a major advantage as it would allow the implementation to be customized according to the typical usage model for the application. The common model involves the tag to be placed next to the reader for a transaction to occur. NonceCrypt can be implemented for both active and passive devices by designing the implementation accordingly. It does not require much more computation or storage space as compared to NFC-SEC-01. A comparison of the memory requirements for the NFC-SEC-01 system and NonceCrypt system is presented in Figure 6.1.

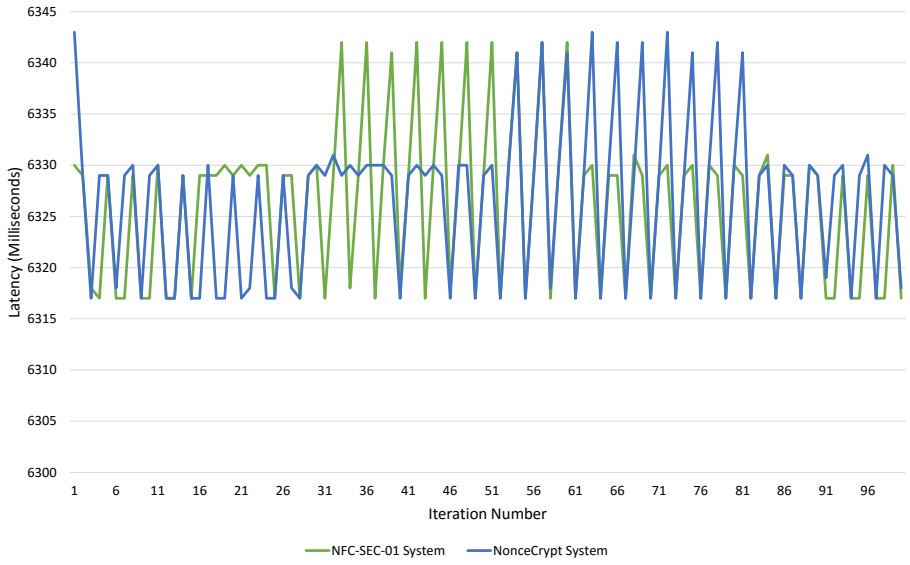

Figure 6.2: Comparison of Latency of the NFC-SEC-01 System and NonceCrypt System.

6.2.4 Overhead

The increase in time overhead caused by computation can be neglected for NonceCrypt. A substantial component of the increase would be caused due to user interaction. In applications which are time sensitive, the implementation can be modeled such that there is no user interaction involved and the protocol is completely automated. But, in case security is a major concern for the application, it would be advisable to include minimal user interaction. Ignoring the time for user interaction, the latency for the NonceCrypt system has been computed to be approximately 6.3 seconds, which is the same as for the NFC-SEC-01 system. The latencies of the NFC-SEC-01 system and NonceCrypt system over 100 iterations of the experiment have been presented in Figure 6.2. Thus, the increase in the time overhead for the NonceCrypt system caused by additional computation is negligible.

6.2.5 Ease of Use

NonceCrypt can be designed as a user-friendly application as the requirements for implementation are very similar to password based systems that exist. Also, as there is a possibility to automate the protocol, it can be made to avoid all user interaction. But, user interaction is a very useful component of security as some extra security may be provided as part of user intelligence. The user might be able to better discriminate between situations that are safe and those that may involve a malicious entity. Hence, the trade-off between security and ease of use is balanced.

7. CONCLUSION

The ubiquitous use of NFC increases the need for better secure standards. A failure in security may lead to loss of private data, possible malicious software being loaded on to the device to compromise its security, and financial frauds. As the risks associated are high, there is a strong need for better security standards to be incorporated into NFC. In the present day, most standards followed by institutions are private secrets and there is a need for a universal set of standards that can be followed.

NFC-SEC provides a security standard that protects against simple attacks, such as eavesdropping, but it does not protect against a more complicated attack such as an MITM attack. As the sensitive data in a transaction can be modified in an MITM attack, this presents a serious drawback. NonceCrypt reduces the scope of attacks that can be carried out by providing protection against MITM attacks and terminating the communication before the data transfer phase, thereby reducing the chance of a malicious entity gaining access to the data. The flexibility of NonceCrypt provides an avenue to tailor the implementation to the purpose of the application. The proposed added layer of encryption will greatly improve the security of NFC transaction at a very small added cost of time and convenience. Hence, NonceCrypt is a robust and general solution.

REFERENCES

[1] Adafruit.com. PN532 NFC/RFID controller breakout board. `https://www.adafruit.com/products/364`. Accessed: 2016-01-31.

[2] Gergely Alpar and Jaap-Henk Hoepman. Avoiding Man-in-the-Middle Attacks When Verifying Public Terminals. In Jan Camenisch, Bruno Crispo, Simone Fischer-Hubner, Ronald Leenes, and Giovanni Russello, editors, *Privacy and Identity Management for Life*, volume 375 of *IFIP Advances in Information and Communication Technology*, pages 261–273. Springer Berlin Heidelberg, 2012.

[3] Android.com. Android | Android Pay. `https://www.android.com/pay/`. Accessed: 2015-11-29.

[4] Apple. ApplePay - Apple. `http://www.apple.com/apple-pay/`. Accessed: 2015-11-29.

[5] Arduino.cc. Arduino - ArduinoBoardUno. `https://www.arduino.cc/en/Main/ArduinoBoardUno`. Accessed: 2016-01-31.

[6] Developer.android.com. NFC Basics | Android Developers. `http://developer.android.com/guide/topics/connectivity/nfc/nfc.html`. Accessed: 2015-12-9.

[7] ECMA. *ECMA-385: NFC-SEC: NFCIP-1 Security Services and Protocol*. ECMA International (European Association for Standardizing Information and Communication Systems), Geneva, Switzerland, Jun 2015.

[8] ECMA. *ECMA-386: NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES*. ECMA International (European Association for Standardizing Information and Communication Systems), Geneva, Switzerland, Jun 2015.

[9] ECMA. *ECMA-409: NFC-SEC-02: NFC-SEC Cryptography Standard using ECDH-256 and AES-GCM*. ECMA International (European Association for Standardizing Information and Communication Systems), Geneva, Switzerland, Jun 2015.

[10] ECMA. *ECMA-410: NFC-SEC-03: NFC-SEC Entity Authentication and Key Agreement using Asymmetric Cryptography*. ECMA International (European Association for Standardizing Information and Communication Systems), Geneva, Switzerland, Jun 2015.

[11] ECMA. *ECMA-411: NFC-SEC-04: NFC-SEC Entity Authentication and Key Agreement using Symmetric Cryptography*. ECMA International (European Association for Standardizing Information and Communication Systems), Geneva, Switzerland, Jun 2015.

[12] Hasoo Eun, Hoonjung Lee, and Heekuck Oh. Conditional privacy preserving security protocol for NFC applications. *Consumer Electronics, IEEE Transactions on*, 59(1):153–160, Feb 2013.

[13] Lishoy Francis, Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. Practical NFC Peer-to-peer Relay Attack Using Mobile Phones. In *Proceedings of the 6th International Conference on Radio Frequency Identification: Security and Privacy Issues*, RFIDSec'10, pages 35–49, Berlin, Heidelberg, 2010. Springer-Verlag.

[14] Lishoy Francis, Gerhard Hancke, Keith Mayes, Konstantinos Markantonakis, and Information Security Group. Practical Relay Attack on Contactless Transactions by Using NFC Mobile Phones. IACR Cryptology ePrint Archive, 2012.

[15] GitHub. adafruit/Adafruit-PN532. `https://github.com/adafruit/Adafruit-PN532`. Accessed: 2016-01-31.

[16] GitHub. kmackay/micro-ecc. https://github.com/kmackay/micro-ecc. Accessed: 2016-01-31.

[17] GitHub. kokke/tiny-AES128-C. https://github.com/kokke/tiny-AES128-C. Accessed: 2016-01-31.

[18] Google.com. Google Wallet. https://www.google.com/wallet/. Accessed: 2015-11-29.

[19] Mohamad Hamze, Fabrice Peyrard, and Emmanuel Conchon. An Improvement of NFC-SEC with Signed Exchanges for an e-Prescription-Based Application. In Gerard Memmi and Ulf Blanke, editors, *Mobile Computing, Applications, and Services*, volume 130 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 166–183. Springer International Publishing, 2014.

[20] Gerhard P. Hancke. Practical Eavesdropping and Skimming Attacks on High-frequency RFID Tokens. *J. Comput. Secur.*, 19(2):259–288, Apr 2011.

[21] G.P. Hancke. A practical relay attack on ISO 14443 proximity cards. Technical report, University of Cambridge Computer Laboratory, 2005.

[22] E. Haselsteiner and K. Breitfub. Security in Near Field Communication (NFC). Strengths and weaknesses. In *In Workshop on RFID Security RFIDSec*, 2006.

[23] Thomas S. Heydt-Benjamin, Daniel V. Bailey, Kevin Fu, Ari Juels, and Tom O'Hare. Vulnerabilities in First-generation RFID-enabled Credit Cards. In *Proceedings of the 11th International Conference on Financial Cryptography and 1st International Conference on Usable Security*, FC'07/USEC'07, pages 2–14, Berlin, Heidelberg, 2007. Springer-Verlag.

[24] ISO 14443. Identification cards - Contactless integrated circuit(s) cards - Proximity cards, 2000.

[25] A. Juels, D. Molnar, and D. Wagner. Security and Privacy Issues in E-passports. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 74–88, Sept 2005.

[26] Z. Kfir and A. Wool. Picking Virtual Pockets using Relay Attacks on Contactless Smartcard. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005.*, pages 47–58, Sept 2005.

[27] Yun-Seok Lee, Eun Kim, and Min-Soo Jung. A NFC based Authentication method for defence of the Man in the Middle Attack. In *3rd International Conference on Computer Science and Information Technology (ICCSIT'2013)*, pages 4–5, 2013.

[28] A. Marcus, G. Davidzon, D. Law, N. Verma, R. Fletcher, A. Khan, and L. Sarmenta. Using NFC-Enabled Mobile Phones for Public Health in Developing Countries. In *Near Field Communication, 2009. NFC '09. First International Workshop on*, pages 30–35, Feb 2009.

[29] C. Mariotti, V. Lakafosis, M.M. Tentzeris, and L. Roselli. An IPv6-enabled wireless shoe-mounted platform for health-monitoring. In *Wireless Sensors and Sensor Networks (WiSNet), 2013 IEEE Topical Conference on*, pages 46–48, Jan 2013.

[30] D.M. Monteiro, J.J.P.C. Rodrigues, and J. Lloret. A secure NFC application for credit transfer among mobile phones. In *Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on*, pages 1–5, May 2012.

[31] C. Mulliner. Vulnerability Analysis and Attacks on NFC-Enabled Mobile Phones. In *Availability, Reliability and Security, 2009. ARES '09. International Conference on*, pages 695–700, March 2009.

[32] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Ramarathnam Venkatesan. Dhwani: Secure Peer-to-peer Acoustic NFC. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 63–74, New York, NY, USA, 2013. ACM.

[33] M.Q. Saeed and C.D. Walter. Off-line NFC Tag Authentication. In *International Conference for Internet Technology And Secured Transactions, 2012*, pages 730–735, Dec 2012.

[34] D. Sethia, D. Gupta, T. Mittal, U. Arora, and H. Saran. NFC based secure mobile healthcare system. In *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on*, pages 1–6, Jan 2014.

[35] E. Strommer, J. Kaartinen, J. Parkka, A. Ylisaukko-oja, and I. Korhonen. Application of Near Field Communication for Health Monitoring in Daily Life. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 3246–3249, Aug 2006.