Security of Near Field Communication: Does My Phone Need A Tinfoil Hat?

Thomas Harren
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
harre096@morris.umn.edu

ABSTRACT

Near Field Communication is a technology that is rapidly growing in popularity and is becoming even more accessible due to the advent of mobile payment systems. *Near Field Communication* (NFC) is built upon High Frequency Radio Frequency Identification technology, more commonly known as HF RFID. However, NFC is a richer communication method: it supports both passive and active components, but at a shorter range. Uses of NFC technologies are being rapidly developed to be used in payment systems and for other applications.

NFC is a flexible communication technology, but it is not inherently secure. The limited range of NFC offers some security, but data transmitted using NFC is still vulnerable to various attacks. As a result, measures to ensure confidentiality, integrity, or authentication need to implemented as an extension of NFC. Moreover, if the data transmitted from a peer is malicious, a hardware-based firewall may be a good way to defend your NFC capable device. One proposed technology is a device-independent security method, a metaphorical tin foil hat, that could offer flexibly against current and evolving attacks on NFC enabled devices.

Keywords

Near Field Communication, Payments, Security

1. INTRODUCTION

Near Field Communication is a technology that is rapidly growing in popularity and is becoming even more accessible due to the advent of mobile payment systems. *Near Field Communication* (NFC) is built upon High Frequency Radio Frequency Identification technology, more commonly known as HF RFID. NFC is restricted to a shorter range than RFID and offers interactive communication method between devices that have both passive and active components. An NFC connection can be set up quickly and connectivity does not require line of sight. Uses of NFC technologies are being rapidly developed to be used in payment systems and in other applications [2].

NFC is a flexible communication technology, but it is not inherently secure. The limited range of NFC offers some security, but data transmitted using NFC is still vulnerable to various attacks. For sensitive data, measures to ensure confidentiality, integrity, or authentication need to implemented as an extension of NFC [4]. In addition, there is no protection against other parties that may have malicious intent. One proposed technology offers a flexible hardware firewall may be an effective way to block data transfers with malicious peers [2].

In this paper, we focus on security and applications of NFC regarding payments and ticketing. First, we describe the foundations of Near Field Communication in Section 2. After the background section, we discuss security in three different NFC contexts: contactless credit cards, mobile ticketing applications, and physical NFC security. In the first souce, we highlight a recent academic source that describes issues and proposes a solution for NFC security in the context of contactless credit cards in Section 3. In Section 4, we discuss a prospective application for the richer NFC communication offered when using mobile phones. In particular, this section focuses on using mobile phones for mass transit ticketing. Three implementations of mobile ticketing, each balancing security and transaction time in a unique way, are introduced, prototyped, and critiqued. This leads to the final source, a discussion about the EnGarde shield in Section 5. Commercial payment systems such as Apple Pay and Android Pay are bringing NFC to mobile phones, which could introduce security risks in both payment and non-payment applications of NFC. The proposed device independent, hardware-based firewall may be a viable way to defend against more general threats.

2. BACKGROUND

In this section, we provide an overview of Near Field Communication and properties of its physical operation. We first discuss RFID technology, the parent technology of NFC. Next, we discuss additional features specific to the NFC standard. Finally, we highlight the need for explicit security when using NFC.

2.1 Elements of HF RFID: Tags & Readers

NFC is a wireless communication standard that is based on, and fully compatible with, the HF RFID (high frequency radio-frequency identification) standard [2]. At a fundamental level, this means that communication happens between

UMM CSci Senior Seminar Conference, April 2016 Morris, MN.

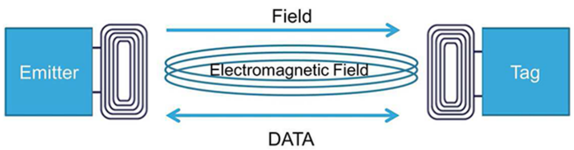

Figure 1: Readers and tags interact with each other using tuned antenna coils and electromagnetic induction. [1]

tags and readers.

A *tag* is composed of an integrated circuit and an antenna. A tag is capable of storing a unique ID and a limited amount of data, which can be read/write or read only. RFID tags can be actively powered, battery assisted, or passively powered. A passive tag relies exclusively on energy induced into the tag's antenna coil. Since passive tags require no built in power source, they are the least expensive and smallest RFID tags. While the tag is powered, it can use its antenna coil to relay its internal information back to the asking party. [8]

A *reader* is a device used to power and interrogate RFID tags. A reader emits an electromagnetic field in order to power nearby tags. Before initiating communication, the reader runs a discovery protocol. If multiple tags respond, the reader uses its collision avoidance protocol to establish communication with a single tag using one tag's unique ID. The tag and the reader then communicate by taking turns sending and receiving messages. [2]

A standard RFID reader-tag interaction is illustrated in Figure 1. Both the reader and tag have antenna coils tuned to 13.56MHz. When the reader generates an electromagnetic field, the reader and the tag are coupled, and power is induced into the tag's antenna coil; this energy transfer is similar to that found in electrical transformers. The tag then converts the AC voltage it receives into DC voltage in order to power the tag's circuit. The electromagnetic field of this frequency is able to power a tag within a range of a few centimeters. According to Gummeson et al, the communication distance can be increased up to 1 meter if larger, higher powered readers are used. [2]

2.2 NFC on Mobile Phones

Near field communication also has features that extend beyond the HF RFID specification. In particular, NFC enabled mobile phone can function in several unique modes:

Phone acting as a reader: In this mode, a mobile phone functions as an RFID tag reader. Touching a phone to a tag mounted to a map, for example, could send the phone a hyperlink to a informational page. [3]

Phone emulating a tag: A mobile phone can also function as if it was a passive tag. This mode can be effectively used even when the phone is not powered, because power is induced by an NFC reader. Mobile payments and other ticketing applications would tend toward this interaction mode. [2]

Phone acting as a peer: When two compatible devices are capable of switching between reader and tag emulation mode, they can communicate directly over NFC in a peer-to-peer manner. Peer-to-peer mode offers the highest communication throughput and can be used to implement stronger security [6] or to coordinate mobile file transfers. [2]

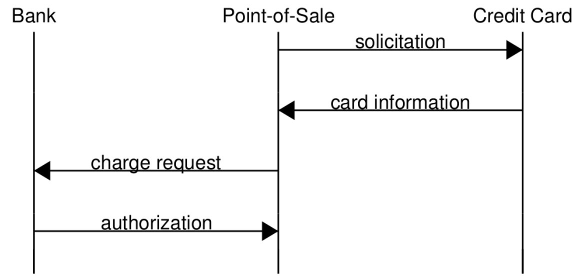

Figure 2: The basic steps executed in an NFC transaction using the current credit card protocol. [4]

2.3 Security for NFC

NFC is a flexible communication technology, but it is not inherently secure. The limited range of NFC offers some security, but data transmitted using NFC is still vulnerable to several attacks. As a result, measures to ensure confidentiality, integrity, or authentication need to implemented as an extension of NFC. Maintaining security of NFC is the main focus of this paper. [4]

3. CONTACTLESS CREDIT CARDS

In this section, we look at the usage of passively powered NFC tags installed into the credit cards and some related security concerns. Jensen, Gouda, and Qiu describe several effective contactless credit cards attacks and propose a security protocol to defend against these attacks. Since their security protocol must run on a passively powered NFC chips embedded in credit card, it is composed of computational inexpensive primitives including pre-computed hashes, indexing and XOR operations. [4]

3.1 Current Credit Card Protocol

As background, we will describe the current credit card protocol used for NFC transactions. The current protocol does offer some level of security by merit of the iCVV. A dynamic card verification value or *iCVV* is single use value that a contactless credit card generates each transaction [7]. The iCVV is returned from a pesudo-random sequence using a seed known only to that specific credit card and the issuing bank. At the time of authorization, the bank checks that the iCVV received is one the expected values in that card's iCVV sequence. The four phases of the credit card protocol used for NFC transactions, are illustrated in Figure 2, and will now be described.

In the first phase, called *solicitation*, the point-of-sale and the credit card exchange several messages in a static manner. In this phase, each both parties share general information about themselves. For example, a card may identify itself as `VISA CREDIT`.

In the second phase, *card information* is sent from the card to the point-of-sale. The card information is composed of the credit card number, the credit card's expiration date, the iCVV, and the name of the bank that issued the card.

The point-of-sale then sends the card information to the bank in the third phase, called the *charge request*. The credit

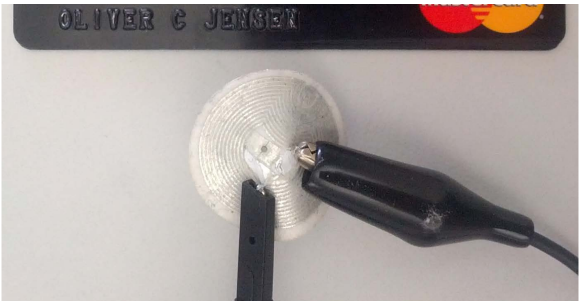

Figure 3: Eavesdropping Antenna [4] (credit card for scale)

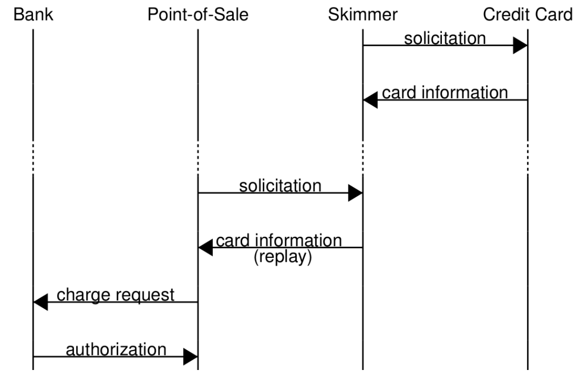

Figure 4: Skimming [4]

card's number and expiration date, along with the iCVV and the dollar amount charged, are sent to the specified bank.

The final phase is called *authorization* and only occurs if the bank deems the card information valid. Banks may also perform other checks based on the transactions physical location or other factors.

3.2 Credit Card Attacks

The following attacks on NFC transactions may vary slightly in agency, but each method ultimately exposes sensitive card information. The first three attacks can be accomplished with merely a NFC compatible mobile phone and some additional, inexpensive hardware. The final attack, compromised point-of-sale, points out a more general weakness about the implementation of the current protocol.

Eavesdropping: In this attack, a malicious party is able to capture sensitive data by listening in on the first two phases of a transaction. Thus, the card number, expiration data, iCVV, and bank name are gleaned by the malicious party. The iCVV cannot be used again, but the other information may already be enough to make a fraudulent purchase.

Jensen, Gouda, and Qiu demonstrated the feasibility of this attack by modifying an NFC tag and connecting it to an inexpensive radio. The very small, easy concealable NFC antenna, shown in Figure 3, that could be mounted or held within the range of a contactless NFC credit card terminal.

Skimming: In this attack, illustrated in Figure 4, a skimmer gains a victims credit card information, including a single usable iCVV, by masquerading as a point-of-sale. After the skimmer has captured this data, it can replay the credit card information to a genuine point-of-sale to perform an illegitimate purchase on behalf of the victim.

Surprisingly, this attack can be carried out by simply installing an Android application called *NFCProxy*.[1] Using *NFCProxy*, any NFC enabled Android device can skim information from a contactless credit card and make a single purchase. To make subsequent purchases, the attack must be repeated to obtain a new iCVVs.

Relay Attacks: The relay attack is similar to the skimming attack, but it does not rely on a single device to skim and replay card data. Instead, two entities work in concert by sharing information over an alternative communication channel, such as wireless LAN.

[1] *NFCProxy* was presented at DefCon 20 and can be downloaded at: https://sourceforge.net/projects/nfcproxy/ [4]

Compromised Point-of-Sale: This attack points out that since point-of-sale devices learn enough information to allow multiple charges, the point-of-sale is a natural target. If a point-of-sale is compromised, transaction becomes accessible to malicious parties. Jensen, Gouda, and Qiu list several merchants that have recently had their point-of-sale systems compromised including Target, Home Depot, and SuperValu stores.

3.3 Proposed Secure Credit Card Protocol

To address these security concerns, Jensen, Gouda, and Qiu have proposed a secure protocol shares the same four phases as the current credit card protocol (Figure 2) but with several variations that will be described and illustrated using the Figure 5.

The solicitation phase now includes a random challenge (**ch**) that will be used by the credit card when building its response. After receiving some basic information and the challenge, the credit card responds by sending the card information in three distinct pieces:

- **A**: ***UUID***, a Universally Unique Identifier that to identify the credit card. The UUID is static.
- **B**: ***H(info, ch, iCVV)*** is used to authenticate the card's identity. Notice that the sensitive info, including the card number and expiration date, will not be transmitted in plain text. The details of the hash-like function H are described below.
- **C**: ***bank name*** is used to route the charge request.

Upon receiving all three parts of the card information, the point-of-sale simply forwards the UUID (A) and the authentication (B) to the bank (C) learning nothing about the actual card data. The challenge (ch) is also sent so that the bank will have all of the pieces necessary to generate the authentication value and check for validity by matching it to B. The bank will also receive the charge amount *$*.

Finally, the bank uses the *UUID* to look up official card data. Then the bank uses the H function with its own copy of the customer information and the challenge (ch) to create B_{bank}. If $B_{bank} = B$, then the bank considers the card data valid and authorizes the charge.

Requirements of function H: So long as B, the value returned from the H function, is indistinguishable from random, no sensitive information can be learned from an eavesdropping attack. Also, when a skimming or relay attacker

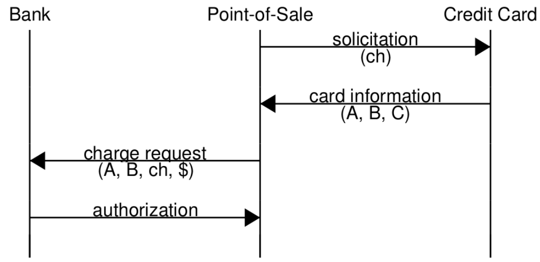

Figure 5: Proposed Credit Card Protocol [4]

collects ch and B and attempts to replay the card information, it must not be able to calculate the new value B' when given the new challenge ch'. Here is the psudo-code for the function H that satisfies these qualities:

```
function G(info, ch):
  const khi = hash(bank_key, info)
  result = empty list of bits
  for each of the n bits of ch:
    if the nth bit of ch is 1:
      append nth bit of khi to result
  return result

function H(info, ch, iCVV):
  x = G(info, ch)
  return (x XOR iCVV)
```

In summary, G composes a binary string x by setting values to 1 at indexes where that value in the challenge matches the value of khi. Note that the value of khi is a constant and need to be computed only when the credit card is manufactured. The value x is then combined with the credit card's freshly generated iCVV using XOR. This creates a value that is indistinguishable from random and cannot be used in replay attacks with other challenge values. Using this method, a low powered credit card can encode and transmit payment information while being protected from all of the aforementioned attacks.

4. NFC AND MASS TRANSIT TICKETING

Mass transit systems have widely adopted contactless NFC cards for identity verification and ticketing. Three Nokia researchers, Tamrakar, Ekberg, and Asokan, have punished a paper investigating the use of NFC-enabled mobile phones for this application. In their paper, they seek to achieve security while keeping transaction time below 300ms – a time budget stated in a white paper by Smart Card Alliance for public transport systems. They first describe the the pieces involved in building a complete identity-verification ticketing architecture. Then, three implementations of mobile ticketing are introduced, prototyped, and critiqued. [6]

4.1 Ticketing Architecture

For NFC phones to be used for mass transit ticketing, additional infrastructure is required. An overview of the high level components and their iterations is illustrated in Figure 6. The Accounting / Certificate Authority (CA) is a centralized entity responsible for issuing transport IDs, clearing transactions and maintaining billing information, and maintaining a blacklist. The User Device (D) is a smartphone capable of executing cryptographic operations. D uses WLAN or a mobile data connection to attain a transport ID from CA. The ticket reader (R) sits at transit gates. R communicates with the D using NFC and the other entities using infrastructure. Finally, the Transport Authority Backend (TA) is an entity capable of operating all of the ticket readers and gates. In addition, the TA collects evidence from D↔R transactions for calculating fares, submits such session evidence to the CA, and queries the CA blacklist in order to distribute changes to all ticket readers.

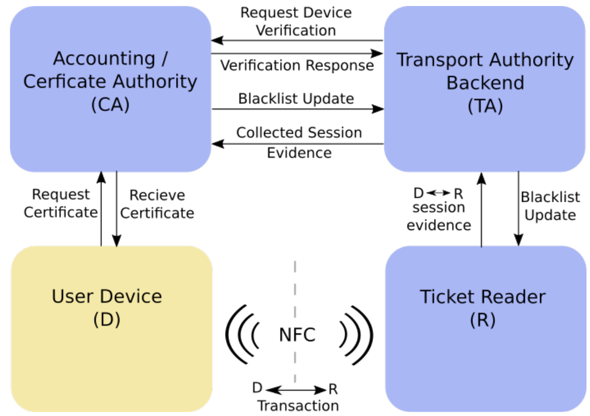

Figure 6: Proposed Ticketing Architecture [6]

The authors note that TA↔CA transactions will not be synchronous to the D↔R data transaction. This way, the ticket lines will not be slowed, with the exception of special blacklist query for user's first transit ride.

4.2 Security Tools

The limited range of NFC offers some security, but data transmitted using NFC is still vulnerable to several attacks. In order to maintain secure communication, several standard security and cryptography tools will be used.

Trusted Hardware: Mobile phones are capable of doing cryptographic operations which are executed in the phone's *trusted execution environment*, or TEE. Several types of TEEs have been developed in the last decade and are now widely deployed on phones. Some TEEs are hardware agnostic while others extend core processing to strengthen security. [6]

Secure Communications: There are various methods for securing communication over an insecure channel such as NFC. In a simple symmetric key system, both parties have a secret key that is used to encrypt and decrypt messages at each end of a communication channel. This works very well, but distributing the private key to both parties can present a problem. The asymmetric key system works differently, requiring a both a private key and a complementary public key. Public keys are distributed and can be used by anyone to encrypt a message that can only be decrypted by the matching private key.

In addition to messages being encrypted, they can also be authenticated and protected against modification using a *signature* or *message authentication code* (MAC). Both of these tools are similar as they use keys to generate a distinct, verifiable message based on the contents of the message body. MACs create this verifible message using a symmetric key, while signatures use an asymmetric key.[5]

4.3 Standard Protocol

In the standard protocol, D first uses WLAN or mobile data to attain an identity *certificate* from CA that is valid for several months. A certificate is a digital document containing attributes associated to the holder, in this case D, from the trusted party, the CA [5]. The standards-compliant X.509 certificate standard is typically used, but its size is too large for this application. Instead, the Nokia researchers have chosen to use an optimized certificate containing an expiration date, a transport identity for D, and a public key to be used by R in decoding messages from D.

Once D has requested and received a certificate from the CA, several steps occur to complete a D↔R transaction. First, R initiates communication, sending its identity and a random challenge to D. D then responds by sending back its certificate. Next, D uses its private key to generate a signature using the challenge along with the IDs of both D and R; D then sends the signature as well. Finally, R validates the certificate using a public key from the CA, verifies the signature, checks D against its blacklist, opens the transit gate, and forwards all transaction to the TA. D also keeps evidence of the transaction which can be used to prove ticket purchases.

The drawback of using a certificate, however, lies in the large keys required by the *RSA* algorithm. RSA is the chosen asymmetric cryptography system and lengthy RSA keys take substantial time to transmit over NFC.

4.4 Variant 1

In Variant 1, the certificate is replaced by a token. The proposed tokens will only be valid for a few hours and will be significantly smaller than the certificate because they do not contain a public key. The token will contain the transport identity for D and two time stamps, forming a window of validity. As a side affect of not passing the public key to the R, Variant 1 cannot use signature. Instead, a MAC, generated using a key shared between D and CA, will be used. Also, since only D and CA know the private key used to generate and check the MAC, R will simply forwards this data to the CA, via the TR for validation. As a result of these changes, the transaction is substantially faster, but validation during a D↔R transaction is limited to R verifying the token and querying the blacklist.

4.5 Variant 2

Variant 2 is built upon Variant 1, but uses a reversed hash chain[2] to build a hybrid system that uses a long term certificate along with timely hash chain information. Before a D↔R transaction begins, D retrieves a certificate, containing the final hash chain term t_f, along with the seed for computing the entire hash chain[3]. The D↔R transaction flows much like the previous protocols using a certificate and a MAC, but the hash chain adds extra security. For the hash chain to work, D sends R a term t and a value n. R can then use t as a hash seed and use the hash function n times. If t and n are correct, the output will match t_f, which was received in the certificate. After a time period (eg 5 minutes), the term t is considered common knowledge and R will only accept a new, unused term. At this point, D will use an earlier term, $t - 1$.

Revealing terms of a hash chain in reverse order is useful because it gives D a series of keys that R can validate, but other parties will not be able to predict. Caveats of this system include the need for D and R to have synchronized clocks and the need to track used hash values in real time. The authors suggest that each time a reader R accepts a valid hash term, that term could sent to the TA and then distributed to all readers to prevent the reuse of hash terms.

4.6 Viability Mobile Phone Ticketing

Based on experimental measurements, Tamrakar, Ekberg, and Asokan, calculated transaction speeds for each protocol at various encryption key sizes. The results, displayed in Table 1, reveal that several of the speeds are very close to or beyond the 300ms threshold. They also note that 1024 bit key size has been deprecated by EMV since 2009 and that 1152 bit keys are stated as acceptable up to 2011. Interestingly, the limited NFC data transfer speeds are the what was found to be the biggest performance bottleneck.

Table 1: Estimated protocol speeds [6]

RSA Key Size	Standard	Variant 1	Variant 2
1024 bits	296 ms	164 ms	182 ms
1152 bits	314 ms	172 ms	190 ms
2048 bits	482 ms	228 ms	246 ms

In terms of security, the Nokia researchers grant that since users need not interact with phone to complete transactions, relay attacks are possible on all protocol variants, especially if the ticketing application is set to always-on to improve usability. Using perishable tokens or the hash chain may mitigate threats, the better user experience and accounting offered by using phones for ticketing may be worth the risk – especially when the risk is small theft in a well-known, guarded transit station. Furthermore, contactless cards, an alternative NFC ticketing technology, does not meet performance or security needs. With these constraints in mind, the researchers contend that, although imperfect, the variant protocols appear be valid paths forward as mobile-based ticketing continues to mature.

5. ENGARDE: A PHYSICAL APPROACH TO NFC SECURITY

Commercial payment systems such as Apple Pay and Android Pay are bringing NFC to our phones, which could introduce security risks in both payment and non-payment applications of NFC. A programmable, hardware-based firewall may be a viable way to defend against more general threats as new applications and attacks are developed. EnGarde is designed to be an extremely power efficient, semi-permanent attachment for everyday mobile phones. Such a device-independent security method, a metaphorical tin foil

[2]A hash chain depends on some hash function that can easily generate term $t + 1$ in the sequence by hashing term t. However, given t and the hash function, $t - 1$ cannot be deduced. When the hash chain is revealed in reverse, a new $t - 1$ can be revealed when desired and easily verified by using the hash function to generate the known term t.

[3]It is unclear if the of hash chain computation occurs on the CA or D in Tamrakar, Ekberg, and Asokan's paper. I am presenting their work with the assumption that the hashing takes place on D.

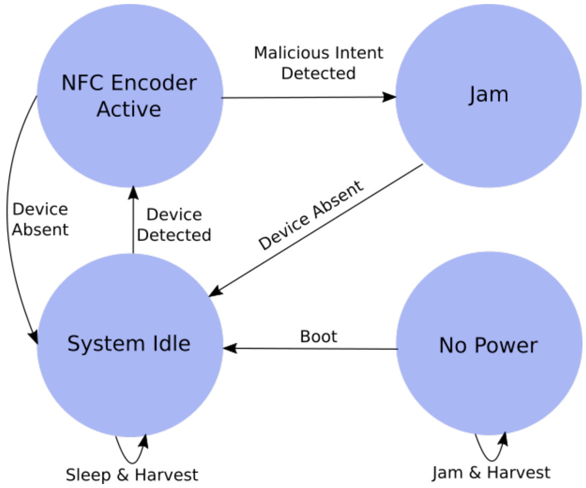

Figure 7: EnGarde[2] state digram

hat, could offer flexibly against current and evolving attacks on NFC enabled devices. [2]

Our goal is to give an overview of the EnGarde security system prototyped by Gummeson et al. Figure 7 is used as a roadmap for our summary of EnGarde and its components.

5.1 No Power Mode

EnGarde is device-independent and is thus not powered directly from the battery contained within the cellphone it is mounted to. Instead, EnGarde contains its own dedicated battery that is charged exclusively by electricity induced into its NFC antenna.[4] As a result of being independently powered, EnGarde can run on battery and encounter a no-power state. Thus, EnGarde was intentionally designed to fail safe. When EnGarde is in the no power state, it cannot do anything until an NFC signal is detected from either the host phone or an external device. At this point, EnGarde collects power from the NFC signal while simultaneously jamming the ongoing communication. Once enough power is collected, control is handed to the microcontroller.

5.2 System Idle Mode

In the system idle state, the microcontroller is running and the EnGarde device simply manages power and waits for an NFC device to move into its vicinity. When an NFC signal is detected, the NFC decoder is activated.

5.3 NFC Decoder Active Mode

In this state, we discuss EnGarde's ability to use discretion to block or allow each NFC communication. To do this EnGarde actively scans each transmission from the nearby NFC device in order to determine the other party's intent. EnGarde is designed to offer real-time protection against malicious attacks from all NFC modes:

Malicious Tags: NFC tags can be handy for storing data or URLs in real world applications such as content rich maps or posters. However, such a tag may contain undesirable content, such as a URL to a malicious website.

Malicious Peer: Since NFC supports file transfers from peers, an NFC phone is ultimately vulnerable to whatever is sent from the peer.

Malicious Reader: Unauthorized readers may attempt to interact with a phone when it is in tag-emulation mode. The phone's location would be known each time a tag ID is read, which effectively enables a form of tracking on that mobile device. Alternatively, a malicious reader could potentially compromise the phone owner's financial data.

Malicious Software Installations: The phone owner may inadvertently install malicious software with permission to broadcast via NFC. EnGarde should be able to prevent undesired information sharing over the NFC interface.

To handle malicious communication, EnGarde scans each message and uses a set of blocking rules to determine if that message should be allowed. The EnGarde is versatile in that current and future undesirable transmissions can be addressed by updating the blocking rules and blacklist.

5.4 Jam Mode

When in this mode, EnGarde's goal is to prevent malicious incoming and outgoing communication over NFC. To do this, EnGarde depends on two jamming primitives:

Reflective Jamming: This defense mechanism is effective against attacks from low-powered tags containing items such as malicious URLs. It works by simply generating a weak signal on the same frequency that the tag is broadcasting to. Since EnGarde is mounted on the back of the owner's phone, EnGarde's signal will be stronger and will effectively block the malicious tag's messages. In additional, the electricity being used to power the tag will also be used to power EnGarde's active defense.

Pulse Jamming: If the phone is being attacked by a powered reader or peer device, a much stronger defense, namely generating a competing active transmission, is required to protect the mobile phone. A continuous active transmission would demand far more power then EnGarde could scavenge. Gummeson et al's response to is to simply corrupt incoming communication in this case. To corrupt the incoming signal, EnGarde needs to generate a pulse lasting only about 20 microseconds. This brief duration is long enough to corrupt two bits of data, even at the slowest NFC transmission rate.

There is, however, a drawback to the pulse jamming method; a sufficiently high-powered reader could generate a strong enough signal to nullify EnGarde's attempts to corrupt the incoming data. Yet, Gummeson et al counter that an active attack from a high powered reader could be mitigated by using the *reflective jamming* method during the offending reader's discovery protocol. If a connection with a high-powered reader is never established, then EnGarde would not have to use the pulse jamming mechanism against a high-powered reader.

5.5 Experimental Evaluation of EnGarde

Jamming: Both of EnGarde's jamming primitives are surprisingly effective. In fact, when Gummeson et al evaluated their device, they found that reflective jamming worked flawlessly against four tags that they tested against. Addition-

[4]Power scavenging methods are addressed in great detail in the primary source, but for the this paper, we will not focus how harvesting adequate power from the cellphone works in practice.

ally, they tested the pulse jamming method with general purpose NFC reader and found that EnGarde was able to block 100% of the responses.

Decoding: Gummeson et al tested EnGarde's decoder using an explicitly defined blacklist set to block all URLs starting with *http://www.malware*. When trying to read tags, one of which contained a blacklisted URL, they found that EnGarde blocked the malicious URL and allowed the benign URL flawlessly.

Based on these results, it appears that the EnGarde hardware can be very effective at blocking NFC communications it deems malicious. We would like to note, however, that EnGarde's ability to block malicious messages is only as good as its ability to detect them; a blacklist containing only *http://www.malware* clearly won't block all NFC communications of malicious intent. EnGarde does, however, offer a programmable, platform independent hardware defense that could certainly be an effective building block for securing NFC is it matures in the future.

6. CONCLUSION

NFC is a young technology and is rapidly growing in popularity due to the deployment of mobile payment system. While the quick setup and ability to communicate with passive tags is attractive, we think that NFC needs to overcome a few hurdles before it becomes ubiquitous. Namely, based on the research we have covered in this paper, security and data transfer speed are significant issues facing NFC. In the first paper, Jensen, Gouda, and Qiu were able to provide a clever solution to the present security holes by introducing a fortified security scheme for contactless credit cards. While their solution is certainly novel, we discovered that the limited range of NFC is not enough to guarantee security. In the second paper, we reviewed several implementations of mass transit ticketing using NFC on mobile phones. The Nokia researchers were able to provide to provide acceptable solutions given the current state of technology, but have clearly highlighted that the slow data transfer speed of NFC is an obstacle to work around. Finally, Gummeson et all introduced a clever, device independent hardware firewall for NFC on an everyday mobile device. The design of the EnGarde device appears to be incredibly successful at jamming signals, but we note that the protection may only be as good as the blacklist it uses. Overall, it appears that NFC is a user friendly, flexible communication interface that may become more popular. However, to gain traction, more clever solutions and faster data transfer speeds will be required in order to add sufficient security in the future.

Acknowledgments

At this point, I would like to extend thanks to everyone who helped me in this project. In particular, my advisor Nic McPhee, along with professors Elena Machkasova and KK Lamberty provided abundant guidance and feedback. I would also like to extend a special thank you to Kevin Byod for his insightful knowledge concerning physics and data transfer protocols.

7. REFERENCES

[1] An introduction to near field communications.

[2] J. J. Gummeson, B. Priyantha, D. Ganesan, D. Thrasher, and P. Zhang. Engarde: Protecting the mobile phone from malicious nfc interactions. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 445–458, New York, NY, USA, 2013. ACM.

[3] R. Hardy, E. Rukzio, P. Holleis, and M. Wagner. Mobile interaction with static and dynamic nfc-based displays. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, pages 123–132, New York, NY, USA, 2010. ACM.

[4] O. Jensen, M. Gouda, and L. Qiu. A secure credit card protocol over nfc. In *Proceedings of the 17th International Conference on Distributed Computing and Networking*, ICDCN '16, pages 32:1–32:9, New York, NY, USA, 2016. ACM.

[5] C. Paar and J. Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[6] S. Tamrakar, J.-E. Ekberg, and N. Asokan. Identity verification schemes for public transport ticketing with nfc phones. In *Proceedings of the Sixth ACM Workshop on Scalable Trusted Computing*, STC '11, pages 37–48, New York, NY, USA, 2011. ACM.

[7] Wikipedia. Card security code — Wikipedia, the free encyclopedia, 2016. [Online; accessed 18-March-2016].

[8] Wikipedia. Radio-frequency identification — Wikipedia, the free encyclopedia, 2016. [Online; accessed 10-March-2016].