Securing NFC Credit Card Payments against Malicious Retailers

Oliver Jensen, Tyler O'Meara, and Mohamed Gouda

University of Texas at Austin

Abstract. The protocol by which "contactless" (NFC) credit cards operate is insecure. Previous work has done much to protect this protocol from malicious third parties, e.g. eavesdroppers, credit card skimmers, etc. However, most of these defenses rely on the retailers being honest, and on their Points of Sale following the credit card protocol faithfully. In this paper, we extend the threat model to include malicious retailers, and remove any restrictions on the operation of their Points of Sale. In particular, we identify two classes of attacks which may be executed by a malicious retailer: Over-charge attacks exploiting victim customers, and Transparent Bridge attacks exploiting victim retailers. We then extend the protocol from previous work in order to defend against these attacks, protecting cardholders and honest retailers from malicious retailers.

1 Introduction

In any credit card purchase, there are two primary parties: the customer and the retailer. Each party controls a device: the customer controls a credit card, and the retailer controls a Point of Sale. It is these devices which communicate on behalf of their controlling parties to coordinate a transaction. The Point of Sale subsequently communicates with the credit card's issuing bank to coordinate the transfer of funds.

Traditional magnetic-stripe credit card readers have been in operation for many years, but they face several important drawbacks: it is easy to accidentally de-magnetize your credit card, and dirty or corroded contacts can make even a well-magnetized card difficult to read. As a result, it is not at all uncommon for a retailer to need to swipe a credit card multiple times before a successful read occurs. Contactless credit card systems solve these problems, using a short-range wireless channel called NFC to communicate with a chip residing within the credit card. This results in more robust credit cards, and less maintenance on credit card readers.

Unfortunately, the protocol used by current NFC credit card payment systems for communication between the Point of Sale and the credit card is insecure. The communications are not encrypted, and the only protection afforded to the customer is the inclusion of a single-use card verification value (called an $iCVV$). This iCVV, freshly generated by the credit card for each transaction, is unpredictable to third parties and thus (in theory) a charge accompanied by a valid

iCVV must have come from the credit card. However, the only thing a valid iCVV assures is that the credit card was, somehow, involved in the process.

Previous work [8] has focused on securing this protocol against malicious third parties (other than the customer and the retailer). It examines four classes of attackers: eavesdroppers, skimmers, relay attackers, and compromised Points of Sale. In all of these attacks, the attacker gains sensitive cardholder information (i.e. the credit card number and expiration date), since the NFC credit card protocol does nothing to conceal it. Skimmers and relay attackers can easily make fraudulent use of credit card data, since by skimming a credit card they also acquire an unused iCVV (rendering this defence nearly useless). The previous work proposes a modification to the NFC credit card protocol, which prevents the abovementioned attacks, with very little additional computation.

In this paper, we extend this protocol to defend against a new class of attacker: the malicious retailer. Traditionally, systems involving an authentication card (e.g. credit cards, building entry, etc.) focus on protecting a system from unauthorized users, but do little to protect users from a malicious system. This assumption is typically justified when the system is a unified entity such as an office building or communal garage. Credit cards break this mould, wherein every retailer is in control of their own device, and it is to these devices that a credit card holder must authenticate. That is, the credit card model implicitly trusts retailers, and the Point of Sale devices under their control.

We make the case that retailers should not be implicitly trusted. We enumerate two attacks which a malicious retailer may perpetrate: a simple over-charge attack, and a more complex "transparent bridge" attack. These attacks both stem from the lack of involvement of the customer in the protocol, and the ability of the retailer's Point of Sale to display one price to the customer, and then charge a different price to that customer's credit card. We build off of the ideas in the Secure CC Protocol [8] and extend it, preventing these attacks.

2 NFC Credit Card Payments

A credit card payment system has five fundamental principals:

1. A **Customer** who wants to make a purchase.
2. A **Bank** at which the Customer has an account.
3. A **Credit Card** issued by the Bank to the Customer.
4. A **Retailer** from whom the Customer wishes to make the purchase.
5. A **Point of Sale** controlled and initialized by the Retailer. It displays the purchase price to the Customer, and communicates with both the Credit Card and its issuing Bank to coordinate the transaction.

It is increasingly popular for retailers to support credit card payments over NFC. NFC is a very attractive channel for use in payment systems, because it provides the benefits of wireless communication, while simultaneously mitigating many of the drawbacks commonly associated with wireless channels:

– NFC is a wireless channel, and thus is unaffected by card demagnetization or read errors due to dirty or corroded contacts.
– NFC has a very short range, mitigating many privacy concerns associated with wireless channels.
– NFC supports communication with unpowered (termed "passive") devices, meaning that a payment device (e.g. a credit card) need not have its own power source.

In an NFC credit card payment system, the Customer indicates an intention to pay by enabling communication between the Point of Sale and his NFC Credit Card. This is done by bringing the Credit Card within range of the Point of Sale (no more than 4 centimiters away). Once within range of each other, the Point of Sale may send messages to the Credit Card and receive any resulting responses.

We will refer to the protocol currently used by NFC credit card payment systems to coordinate a transactions as the "Original CC Protocol". The messages involved in this protocol, illustrated in Figure 1, are as follows:

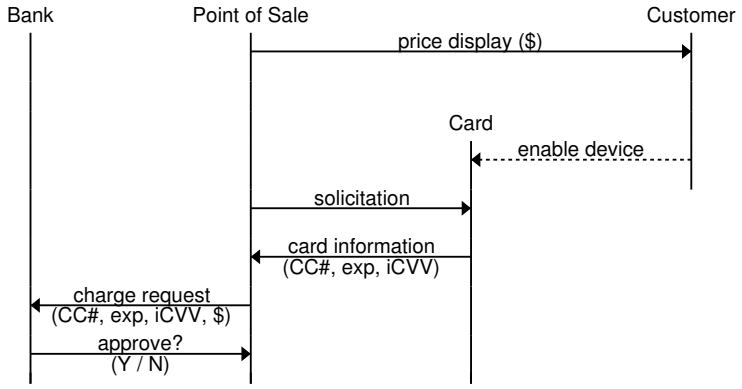
1. The Point of Sale displays the price of the purchase on its screen, while simultaneously attempting to establish communication over NFC.
2. If the Customer agrees with the displayed price, he brings his Credit Card within 4 centimeters of the Point of Sale and communication between the Point of Sale and the Credit Card is established.
3. The Point of Sale sends a *solicitation* message to the Credit Card.
4. The Credit Card responds to the solicitation message with a *card information* message, supplying the Point of Sale with the necessary information to issue a charge, and identifying the Credit Card's issuing bank.
5. Then the Point of Sale sends a *charge request* message to the Bank. This message is sent securely over the Internet.
6. The Bank verifies the details of the charge request, and responds to the Point of Sale with a *acceptance* message, indicating whether the charge has been accepted.

The message contents in the Original CC Protocol are as follows:

Solicitation: In practice, the solicitation message actually consists of a number of messages sent in both directions. Its purpose is to exchange information about the Credit Card type (e.g. *Visa Credit*) and the Point of Sale model (e.g. *2PAY.SYS.DDF01*), which defines the format of subsequent messages. It is a choreographed dance with a specific (and constant) set of messages for a given model of Point of Sale and Credit Card, so we abstract this conversation to a single solicitation message.

Card Information: This message contains all information necessary to coordinate an arbitrary charge request to a credit card's issuing bank. It consists of four components:
  – The Credit Card number, identical to the number printed on the front of the card.
  – The Credit Card's expiration date.

Fig. 1. The Original CC Protocol



- An *iCVV* ("integrated Card Verification Value"). This iCVV is a security code, similar to the 3-digit number printed on the back of a credit card, but is newly generated for each transaction. It is an element in a pseudo-random sequence generated by a secret seed known only to the Credit Card and its issuing Bank, making it unpredictable to third parties.
- The issuing Bank name. This is used for routing purposes, and is not a component of the subsequent charge request. As such, it is not pictured in Figure 1.

Charge Request: This message is sent to the Bank identified in the card information message, and consists of four components:

- The Credit Card number, identifying the account to be charged.
- The Credit Card's expiration date.
- The Credit Card's iCVV.
- The dollar amount to be charged.

Approval: This message consists of a *response code* determined by the Bank, indicating its decision relating to the charge request. The bank makes this decision after verifying the information supplied in the charge request, and performing additional checks such as matching the purchase to a known location of the Customer. The most common response codes are the result of a simple approval decision (i.e. "Approved" or "Declined"), although a number of different codes (e.g. "Pick up card" if the card was reported lost or stolen, "Waiting for line" to indicate that the issuer's lines are currently busy) are supported. We abstract this message as a single bit: whether or not the Customer's account has been charged.

3 Defending Against Malicious Third Parties

While the iCVV in the Original CC Protocol does offer some protection from fake credit card charges by ensuring that the credit card was involved in some

way, there is much that it cannot defend against. As discussed in [8], the Original CC Protocol is vulnerable to four types of attacks that can be launched by a malicious third party (an entity separate from the Customer or Retailer). These four types of attacks are:

Eavesdropping: wherein a malicious third party listens in on a transaction to learn the Credit Card's number and expiration date.

Skimming: wherein a malicious third party harvests payment information from a Credit Card, and then uses it to perform a fraudulent purchase.

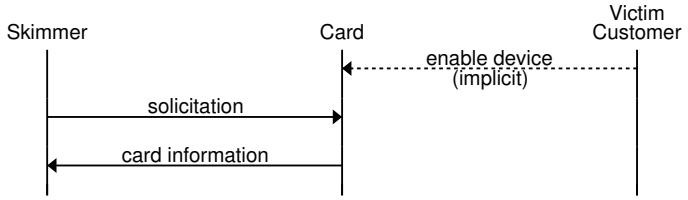
Relay attacks: wherein two malicious accomplices use skimming-like behavior and out-of-band communication to connect a Credit Card to a Point of Sale well beyond NFC range.

Attacks facilitated by a compromised point of sale: wherein a malicious third party has actually compromised the Retailer's systems, harvesting credit card information.

To illustrate these attacks, we will discuss the skimming attack, launched by a malicious third party, called the *Skimmer*. The Skimmer controls an NFC-capable smart phone, and approaches an unsuspecting Credit Card to be within NFC range. He then uses his phone to impersonate a Point of Sale to the Credit Card, soliciting the card for its payment information. This impersonation is not difficult as there is no authorization taking place – indeed, an Android application called NFC Proxy [1] exists to make this attack trivial to execute.

This attack is illustrated in Figure 2.

Fig. 2. A Skimming Attack



Skimming a credit card does not require explicit authorization from the card's owner: an attacker needs only to bring their phone within range of the victim's pocket to communicate with an NFC credit card, as a Credit Card assumes that being able to receive a solicitation message is tantamount to the Customer intending to make a purchase. A fleeting proximity between the Skimmer's device and the Credit Card, perhaps standing in line at a coffee shop or on a crowded subway, is all that is needed. A fraction of a second suffices.

The Original CC Protocol also implicitly trusts the ability of a Retailer to keep its data secure. By allowing persistent sensitive information (e.g. the credit card number and expiration date) to be transmitted to a device under the Retailer's control, this protocol invites attacks on the Retailer's own systems. This

is a very real threat, as evidenced by recent events: over the last three years, a number of high-profile attacks against chains such as Target, Home Depot, Nieman Marcus and P.F. Chang's have delivered *hundreds of millions* of credit card records into the hands of attackers [14] [15] [4] [9] [11]. These records consist of credit card numbers with expiration dates, and in many cases also the cardholder names, billing addresses, and any other information the retailer may have access to.

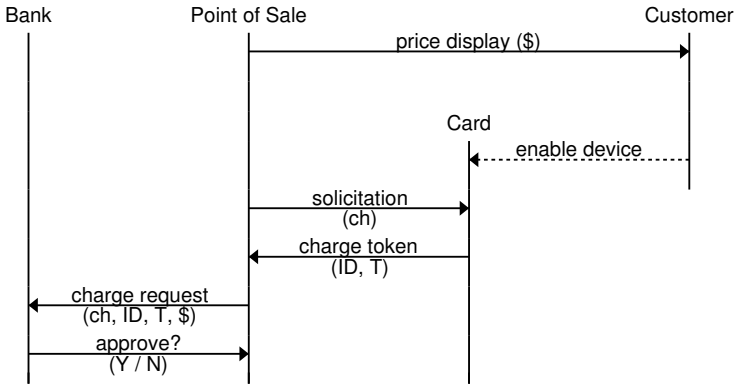
Previous work in [8] has described simple and inexpensive ways to thwart these attacks and others, proposing a replacement protocol termed the "Secure CC Protocol". When a credit card is solicited under the Secure CC Protocol, the Point of Sale includes a randomly generated challenge value *ch*. Instead of responding with the card's private information (i.e. the credit card number and expiration date), the Credit Card transmits its card *ID* (a unique identifier, not considered private) accompanied with a token T, valid for a single purchase. An outline of this protocol is shown in Figure 3. Its messages consist of the following:

1. The Point of Sale displays a price on its screen as before, prompting the customer to bring his credit card within NFC range of the Point of Sale.
2. The Point of Sale sends a *solicitation* message to the credit card, including a random challenge *ch*.
3. The credit card responds with an (*ID,* T) pair accompanied with the issuing bank's name for routing. *ID* is a universally unique identifier (also known as a UUID or GUID) to identify the card. While this value can be used to "track" the card, it is not considered sensitive as it serves no other purpose. T is a token which authorizes a single purchase for a given challenge. The function by which it is generated can be thought of as a function which concatenates the challenge *ch*, a card specific secret value known only to the card and the bank, and the iCVV, and then hashing the result.
4. The Point of Sale sends a charge request to the bank consisting of the (*ch, ID,* T) tuple, accompanied with the dollar amount that the retailer wishes to charge.
5. The bank looks up the account associated with *ID* in order to ascertain the card-specific secret value and the next expected iCVV. It then calculates $T_{bank} = Hash(ch, secret, iCVV)$, verifying that $T_{bank} = T$ to authenticate the charge. The token T being dependent on the challenge *ch* renders skimming and similar attacks impotent, since the attacker cannot predict the challenge which it will be issued when attempting to use skimmed data.

4 Malicious Retailers

Previous work (including the Secure CC Protocol [8]) has focused primarily on defending the Retailer and Customer from malicious third parties, such as eavesdroppers and credit card skimmers. By contrast, we examine the problems posed by malicious retailers, and focus on how to secure NFC credit card payment systems against them. As will be described shortly, attacks by malicious Retailers

Fig. 3. The Secure CC Protocol



are particularly pernicious, as they can be less easily identified as fraud. Even when these attacks are detected, the resolutions are not always simple.

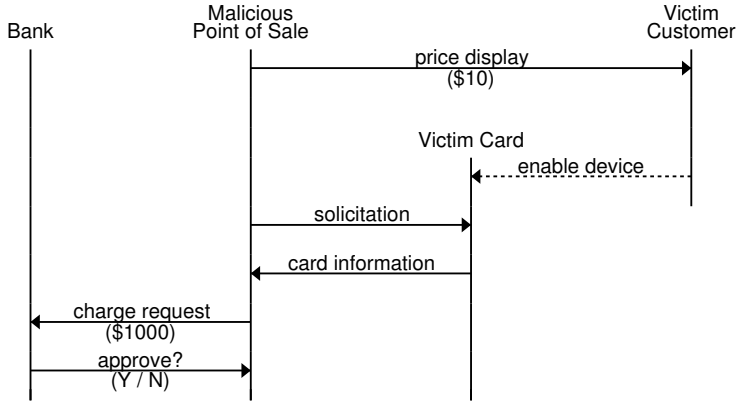
Recall that when making a payment, the Customer first views the price about to be charged on the screen of the Retailer's Point of Sale. Using this information, he makes his one and only decision: to allow the payment protocol to occur, or not. The underlying assumption that the customer makes is that the price displayed on the screen is equal to the price which will be charged to his Credit Card. This need not be the case: the information displayed on a screen is merely an assurance in the informal sense: the numbers displayed to the customer *should* reflect the dollar amount which will subsequently be sent with the charge request, but there is no mechanism in place to require this. As a result, two attacks emerge.

4.1 The Over-charge Attack

An Over-charge attack is characterized by the malicious Point of Sale displaying one price to the customer (in the *price display* message of the CC Protocols shown in Figures 1 and 3) and then sending a higher price to the Bank (in the *charge request* message of the CC Protocols). As a result, the Customer believes himself to have been charged one amount, but is instead charged an arbitrarily higher amount. Since the Customer is uninvolved in the protocol besides the initial step of allowing it to occur, there is no mechanism ensuring that the price displayed to the Customer matches the price that the (malicious) Point of Sale sends to the Bank.

Should a Customer become aware of an over-charge when reviewing his monthly statement, he may file a charge-back request with his Bank, nullifying the payment as fraudulent. As a result, while the amount by which the Customer may be overcharged is unconstrained by the protocol, it should be relatively small for the attack to ultimately be successful. For example, it is easy to notice a gas station charge for $500.00 instead of $21.87 on a monthly

Fig. 4. Over-charge Attack



statement, and the resulting investigation would be uncomplicated. However, should the struggling business choose to increase charges by 5%, the resulting gas station charge of $22.96 could very easily be overlooked. Even were it to be noticed, the victim Customer may have difficulty proving the discrepancy.

4.2 The Transparent Bridge Attack

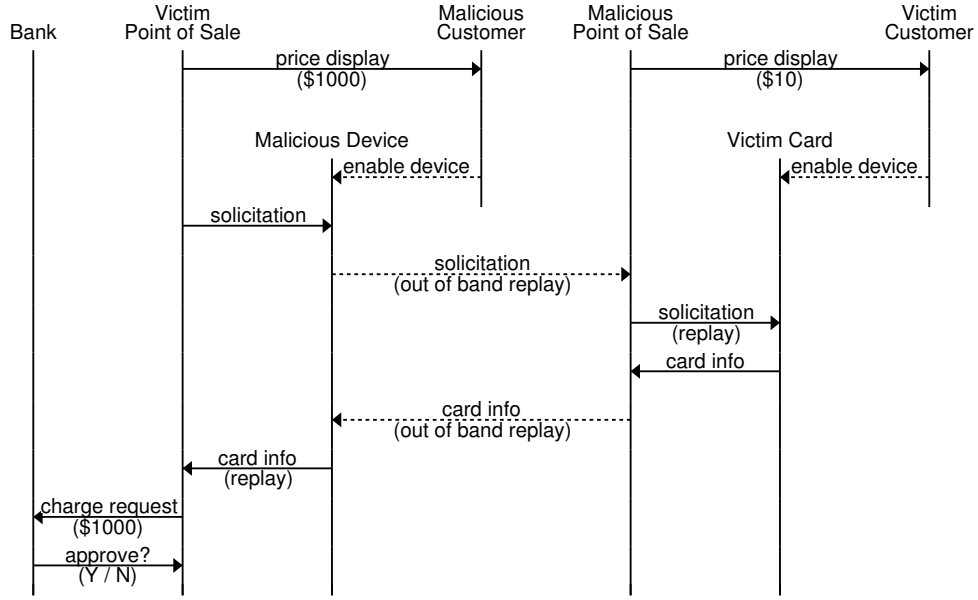
A more interesting attack is described by Drimer and Murdoch [3]. It considers a man-in-the-middle attack, perpetrated by a malicious Retailer and an accomplice with specialized equipment. This attack involves four parties: a victim Customer, a malicious Retailer, a malicious Customer, and a victim Retailer. The malicious retailer and the malicious customer collude to perform this attack.

The malicious Customer is issued with a special card, capable of relaying all messages it receives from a Point of Sale to the malicious Retailer in real time. Similarly, it can relay any responses it receives from the malicious Retailer back to this Point of Sale. As a result, the malicious Customer and malicious Retailer can together form a bridge between the victim Credit Card and the victim Retailer's Point of Sale. The attack is illustrated in Figure 5 and runs as follows:

1. First, the victim Customer attempts to make a relatively inexpensive purchase from the malicious Retailer. Simultaneously, the malicious Customer prepares to make a relatively expensive purchase from a victim Retailer.
2. The victim Retailer's Point of Sale issues a *solicitation* message to the malicious Customer, who relays it to the malicious Retailer.
3. The malicious Retailer then forwards this *solicitation* to the victim Credit Card.
4. The victim Credit Card responds with a *card information* message to the malicious Point of Sale, who relays it to the malicious Customer.

5. The malicious Customer forwards this *card information* message to the victim Retailer's Point of Sale.
6. The victim Retailer issues a *charge request* message to the victim Credit Card's bank, charging the victim Customer for the expensive purchase.

Fig. 5. Transparent Bridge Attack



In this attack, all messages are transparently relayed between the victim Retailer's Point of Sale and the victim Customer's Credit Card. As a result, the victim Customer believes himself to be making an inexpensive purchase at the malicious Retailer, while he is actually making an expensive purchase at the victim Retailer. The malicious Retailer loses the inexpensive sale, but acquires the merchandise from an expensive purchase in exchange.

The Transparent Bridge attack is particularly interesting, because the malicious parties leave no trace with either of the victims: to the victim Customer there is only a record of an expensive purchase at the victim Retailer, and to the victim Retailer there is only the customer record of the victim Customer. The amount which can be successfully stolen by the malicious Retailer is unconstrained, and needs not evade notice: if the discrepancy is noticed and the victim Customer files a charge-back request, it will be against the victim Retailer (and not the malicious Retailer). As such, detected or not, it is one of the two victims that will be left facing the bill, making the Transparent Bridge attack significantly more dangerous than the Over-charge attack described earlier.

Drimer et al. propose a defense against this attack in the context of EMV credit cards (colloquially known as "chip and pin"). However, this solution is not applicable to contactless credit cards, and as such the problem remains open.

5 Defending Against Malicious Retailers

Passive "smart cards" (such as NFC credit cards) are designed primarily to authenticate the cardholder to the system, and not to provide any assurance to the cardholder *about* the system. As a result, they offer little by way of possibility to defend against malicious retailers. However, it has become increasingly common for the devices engaging in NFC credit card payments to break this mould by not being credit cards at all: smart phones with NFC capabilities have given customers the ability (through applications like Android Pay or Apple Pay) to use their phones to emulate a credit card.

This is particularly attractive to many customers, since it allows for the convenience of carrying a potentially unlimited number of credit cards without a bulky wallet, while also affording additional security against theft (by way of passwords or PINs). In addition, such "virtual credit cards" present a rich interface, allowing for finer-grained control and, as a result, stronger defenses against malicious retailers.

Since the aforementioned attacks allowing a malicious Retailer to exploit a Customer are tied to the Retailer's ability to display one price and charge another, our proposed defense against these attacks is built around removing this ability when possible. When using a virtual Credit Card as implemented on a smart phone, the phone's interface provides an additional communication channel between the Customer and the (virtual) Credit Card. This communication channel can be harnessed to allow the Customer to participate in the payment protocol, beyond simply allowing it to occur.

Previous work in the Secure CC Protocol defines a function H, proves several of its properties and uses it to defend against third party attacks like skimmers and eavesdroppers [8]. We note that each property required of this function H is a property enjoyed by common cryptographic hash functions, such as those in the *SHA* family. As such, using a hash function instead of the derived function H does not reduce the security of the Secure CC Protocol.

We propose an extension to the Secure CC Protocol, while altering it to use a cryptographic hash function for simplicity.

5.1 The Extended Secure CC Protocol

In our description of the Extended Secure CC Protocol, we will use the following notation:

ch: a fresh, randomly generated challenge value, chosen by the Point of Sale.
INFO: the Credit Card's payment information, consisting of the Credit Card number and expiration date.

ID: a UUID, uniquely identifying an individual Credit Card without revealing any information about *INFO*.

iCVV: an unpredictable value freshly generated by the Credit Card for each transaction (the issuing bank can generate the same sequence of values).

B: the name of the issuing Bank, used for the purpose of routing transactions as before.

The Extended Secure CC Protocol, operating between a Point of Sale and a virtual Credit Card, is illustrated in Figure 6 and proceeds as follows:

1. The Point of Sale displays a price **$d** on its screen, inviting the Customer to bring his Credit Card within NFC range.
2. The Point of Sale sends a solicitation to the Credit Card, including a fresh random challenge **ch** and the price to be charged **$c**. (Recall that if the Point of Sale is honest, **$c** = **$d**)
3. The virtual Credit Card displays the price **$c** to the Customer, who can choose to accept or reject it. Rejecting the price aborts the protocol here.
4. If granted authorization by the Customer, the Card calculates

$$T = H(\text{INFO}, \text{ch}, \$\text{c}, \text{iCVV})$$

   and responds to the Point of Sale with a card information message consisting of [**ID**, **T**, **B**].
5. The Point of Sale sends a charge request message to the issuing Bank (identified by **B**) consisting of [**ID**, **T**, **ch**, **$r**]. (Again, if the Point of Sale is honest, **$r** = **$d**)
6. The bank uses **ID** to look up $\text{INFO}_{\text{bank}}$ and then calculates $\text{iCVV}_{\text{bank}}$. It then uses the **ch** and **$r** supplied in the charge request message to determine

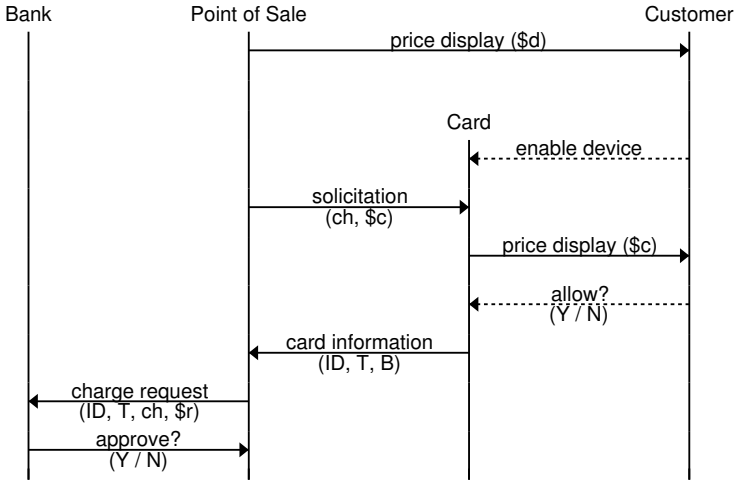
$$T_{\text{bank}} = H(\text{INFO}_{\text{bank}}, \text{ch}, \$\text{r}, \text{iCVV}_{\text{bank}})$$

   If $T \neq T_{\text{bank}}$, the bank will decline the charge, otherwise it approves the charge for **$r**.

When using a physical Credit Card instead of a virtual one, no communication channel exists between the Card and the Customer. As a result, the steps above in which the Card displays the charge price ($c) to the Customer and awaits authorization from the Customer cannot occur. Instead, a physical Credit Card must implicitly assume successful authorization from the Customer, effectively skipping step 3. As a result, while not providing protections from malicious Retailers to physical Credit Cards, the protocol maintains backwards compatibility with no loss of functionality or security against malicious third parties.

We note that a naive implementation of the protocol above might require excessively long timeouts between the Point of Sale sending its solicitation message and receiving the response. Should long timeouts not be desired, a simple solution would be for the Point of Sale to send periodic solicitations (with new

Fig. 6. Extended Secure CC Protocol



challenges). The virtual Credit Card, upon receiving permission from the Customer, could then cache this approval and respond immediately to the subsequent solicitation. Besides noting this particular case, we emphasize that issues such as these are implementation details, the decisions for which are best left to those implementing the protocol.

5.2 Defending Against the Over-charge Attack

The extended protocol prevents the Over-charge attack against Customers using a virtual Credit Card. In step 3, the Customer verifies that $d = $c through visual comparison. Due to theinclusion of $c in the hash when generating token T, we gain the assurance that for any charge accepted by the Bank, $c = $r.

Thus, through a transitive argument, the Customer can be assured that for any successful charge, $d = $r. Should the malicious retailer attempt to issue a charge request with some $r $\neq$ $d, then $T_{\text{bank}} \neq T$ and the charge will be declined by the Bank.

5.3 Defending Against the Transparent Bridge Attack

The extended protocol makes no attempt to prevent this attack from occurring. Instead, it removes the economic incentive of performing such an attack against Customers using virtual Credit Cards.

In the Transparent Bridge attack, the malicious Retailer loses the sale paid by the victim Customer, in return for acquiring the purchase made by the malicious Customer. In order for the Transparent Bridge attack to be viable, the malicious actors must have something to gain: the value of the malicious Customer's purchase must be greater than the value of the victim Customer's purchase. When the extended protocol is used, one of two scenarios occurs:

1. The price associated with the malicious Customer's purchase differs from (i.e. is greater than) the price of the victim Customer's purchase. The victim Customer compares the price displayed by the Point of Sale and the price displayed by his virtual Credit Card. The would-be victim Customer immediately detects the attack and aborts the transaction.
2. The price associated with the malicious Customer's purchase is equal to the price of the victim Customer's purchase. The victim Customer does not detect this attack, and allows the transaction to occur. The end result: the victim Customer paid for the price of what he received, and the victim Retailer received the price of what it sold.

As a result, there is no longer any incentive to carrying out this attack, as the only successful instance results in all parties getting paid exactly as much as they would had they been honest.

6 Related Work

Our work builds primarily upon the Secure Credit Card Protocol over NFC [8], extending the protocol to defend against malicious retailers in addition to malicious third parties.

Kortvedt explores the problem of eavesdropping on NFC communications [10], and suggests a symmetric encryption solution with a strong mutual authentication. Madlmayr et al. analyze the state of NFC communication privacy [13], proposing several technical defenses to threats. Both works [10] and [13] focus on protecting the NFC channel itself, and do not take protocols or applications into account. As a result, while they are effective in defending against channel attacks such eavesdropping, they cannot not affect skimmers, relay attackers, compromised points of sale, or malicious retailers. As such, they fall short of protecting NFC credit card payments.

Haselsteine and Breitfuß provide a broad survey in [7] of several classes of attacks and defenses applicable to the NFC channel. Similarly to [13] and [10], they focus on securing the channel itself from attackers, suggesting that NFC participants perform a key-exchange protocol such as Diffie-Helmann [2], then use this derived secret key to establish a secure channel. As a result, this approach also falls short of protecting NFC credit card payments, for the same reason.

Drimer and Murdoch [3] present an attack on credit card payment systems, which we described in Section 4 as the Transparent Bridge attack. This attack relies on the ability to perform out-of-band real-time proxying and relaying of messages between two parties. Drimer et al. implement this attack against EMV ("chip and pin") credit cards, demonstrating its practicality. They recommend defending against such attacks via distance bounding, essentially measuring round-trip communication timing to detect any delays introduced through the relaying of messages. Such a defense is reasonable when reading responses directly from chip I/O (as in EMV credit card transactions), but does not lend itself well to responses generated by a multitasking computational device such as a smart phone, where delays can be variable depending on unrelated software.

In [6], Francis et al. find that out-of-band real-time proxying and relaying of messages is possible over NFC, constructing a transparent bridge between two NFC devices over Bluetooth. While Drimer et al. only demonstrated the Transparent Bridge attack with EMV credit cards, this result indicates that the attack applies to contactless credit cards as well. Francis et al. propose to use location information such as GPS coordinates in order to detect and defend against this relaying of messages, which in turn would render the Transparent Bridge attack infeasible. However, location information can be unreliable or unavailable in some areas, and as such, one cannot rely on its availability and correctness.

By contrast to [6] and [3], our approach does not seek to detect or prevent attacks relying on the proxying or relaying of information, choosing instead to render them impotent.

In [12], Lee provides some analysis of relay and skimming attacks on NFC credit card transactions, and presents the Android application *NFCProxy* [1] which implements these attacks. This work focuses on demonstrating how easy it is for any would-be fraudster to perform skimming and relay attacks, but does not discuss or propose any countermeasures.

In [5], Eun et al. explore the issue of privacy in the face of NFC eavesdroppers, considering mobile payments as a case study. Not constraining themselves to supporting physical credit cards, they suggest the creation of an "NFC-SEC" protocol complete with key-exchange and public key cryptography. Their approach includes the concept of "unlinkability" (explicitly excluded by the Secure Credit Card Protocol) wherein a merchant cannot correlate multiple purchases from the same credit card.

7 Concluding Remarks

In this paper, we discussed how to extend the Secure Credit Card Protocol over NFC to defend against Over-charge and Transparent Bridge attacks, protecting honest card-holders using virtual credit cards (such as a smart phone running Android Pay) from malicious retailers. The proposed extension, although effective, is simple and computationally inexpensive. It consists of three components:

1. The Point of Sale includes the price of the transaction in its solicitation message.
2. The virtual Credit Card requests confirmation of the price from its card holder before continuing.
3. The Charge Token generated by the card is bound to the confirmed transaction price.

Note that this defense against malicious retailers is only effective when the customer is using a virtual credit card, since physical cards cannot confirm the transaction price with their card-holders. Customers using physical credit cards can still participate in this extended protocol, and will still enjoy the protections against malicious third parties afforded by the Secure Credit Card Protocol. However, defending against the Over-charge and Transparent Bridge attacks

remains an open problem with customers who use physical credit cards rather than smart-devices.

8 Acknowledgments

Research of Mohamed Gouda is supported in part by the NSF award #1440035.

References

1. BlackwingHQ: Nfcproxy. http://sourceforge.net/projects/nfcproxy/ (2012)
2. Diffie, W., Hellman, M.: New directions in cryptography. Information Theory, IEEE Transactions on 22(6), 644–654 (Nov 1976)
3. Drimer, S., Murdoch, S.J.: Keep your enemies close: Distance bounding against smartcard relay attacks. In: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium. pp. 7:1–7:16. SS'07, USENIX Association, Berkeley, CA, USA (2007), http://dl.acm.org/citation.cfm?id=1362903.1362910
4. Elizabeth Harris, Nicole Perlroth, N.P.: Neiman marcus data breach worse than first said. http://www.nytimes.com/2014/01/24/business/neiman-marcus-breach-affected-1-1-million-cards.html, accessed: 2014-11-10
5. Eun, H., Lee, H., Oh, H.: Conditional privacy preserving security protocol for nfc applications. Consumer Electronics, IEEE Transactions on 59(1), 153–160 (2013)
6. Francis, L., Hancke, G., Mayes, K., Markantonakis, K.: Practical nfc peer-to-peer relay attack using mobile phones. In: Radio Frequency Identification: Security and Privacy Issues, pp. 35–49. Springer (2010)
7. Haselsteiner, E., Breitfuß, K.: Security in near field communication (nfc). In: Workshop on RFID security. pp. 12–14 (2006)
8. Jensen, O., Gouda, M., Qiu, L.: A secure credit card protocol over nfc. In: Chan, M.C., Pandurangan, G. (eds.) International Conference on Distributed Computing and Networking. ACM (Jan 2016)
9. Kennedy, C.: Millions of card numbers likely stolen during supervalu data breach, security expert says. http://www.bizjournals.com/twincities/news/2014/08/18/supervalu-millions-card-numbers-likely-stolen.html?page=all, accessed: 2014-11-10
10. Kortvedt, H.S.: Securing Near Field Communication. Master's thesis, Norwegian University of Science and Technology, Norway (2009)
11. Krebs, B.: P.f. changs breach likely began in sept. 2013. http://krebsonsecurity.com/2014/06/p-f-changs-breach-likely-began-in-sept-2013/, accessed: 2014-11-10
12. Lee, E.: Nfc hacking: The easy way. In: Defcon hacking conference. vol. 20 (2012)
13. Madlmayr, G., Langer, J., Kantner, C., Scharinger, J.: Nfc devices: Security and privacy. In: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on. pp. 642–647. IEEE (2008)
14. Robin Sidel, Danny Yadron, S.G.: Target hit by credit-card breach. http://online.wsj.com/articles/SB10001424052702304773104579266743230242538, accessed: 2014-11-10
15. Sidel, R.: Home depot's 56 million card breach bigger than target's. http://online.wsj.com/articles/home-depot-breach-bigger-than-targets-1411073571, accessed: 2014-11-10