NFC-enabled Attack on Cyber Physical Systems: A Practical Case Study

Fan Dang[1], Pengfei Zhou[1,2], Zhenhua Li[1], Yunhao Liu[1]
[1] School of Software, TNLIST, and KLISS MoE, Tsinghua University, China
[2] Beijing Feifanshi Technology Co., Ltd., China
dangf13@mails.tsinghua.edu.cn, {zhoupf05, lizhenhua1983, yunhao}@tsinghua.edu.cn

***Abstract*—Automated fare collection (AFC) systems have been widely applied to practical transportation due to their convenience. Although there are many potential threats of NFC such as eavesdropping, data modification, and relay attacks, NFC based AFC systems are considered secure, due to the limited $10cm$ communication distance. Nevertheless, the proliferation of NFC-equipped mobile phones make such system venerable. We introduce and implement an attack on AFC cards that permits an attacker to top up his smart card and get a refund. We also propose possible countermeasures to defend against these attacks.**

I. Introduction

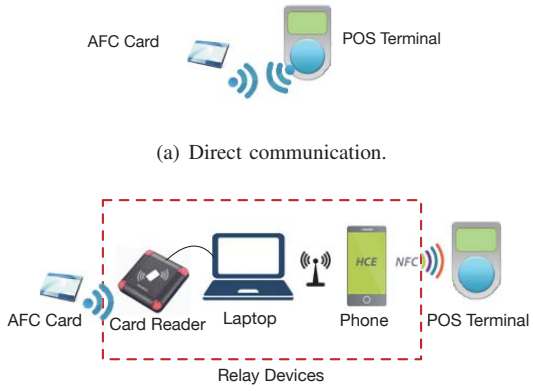
Automated Fare Collection (AFC) systems have been widely applied for decades to automate manual ticketing. They are deployed in major cities all over the world, with billions of contactless smart cards issued. Such a card is able to store and process data, and transceive data with a terminal wirelessly. Thus it is commonly used as an electronic ticket in AFC systems.

The MIFARE chip was developed as a solution for AFC, and it was seen as the major candidate for AFC systems after its introduction for ten years. In 2008, however, researchers discovered a serious security flaw in MIFARE Classic cards [1]–[3]. In particular, the cipher algorithm used in MIFARE Classic, known as CRYPTO1, has been reversed and reconstructed in detail, and a straightforward method can retrieve cryptographic keys. For higher security consideration, consequently, AFC cards are then migrated to processor cards, which employ more secure identity authentication mechanisms. Today, most issued smart cards are processor cards.

Since a huge number of smart cards are in use, even we assume $1 is spent every day on each card, more than 100 billion US dollars can be spent in a year in public transportation systems. As a result, even a tiny flaw may lead to an extremely huge loss.

A symmetric encryption method (*e.g.,* based on 3DES) is commonly used in the smart card system to authenticate the validity during the communication. Besides authentication, however, data are mostly in plaintext [4]–[8]. Such insecurity has been considered acceptable since the AFC network is typically isolated from the public Internet, and the attackers will need to hack into the infrastructure of AFC systems, which is rather difficult.

In 2013, Android 4.4 introduces a special method for card emulation, called host-based card emulation (HCE). The method allows any Android application to emulate a card and communicate directly to a card reader. HCE makes it more convenient for consumers to pay with any NFC Android phones, but also changes the threat landscape. Even there are potential threats of NFC such as eavesdropping, data modification, and relay attacks, as discussed in Section II, NFC was still considered secure, due to the communication distance, which is limited to up to $10cm$. The attacks either simply work in theory, or require specialized equipment. Unfortunately, the advent of NFC-equipped mobile phones bridged the gap between the AFC network and the Internet, thus putting AFC systems in a highly dangerous situation.



(a) Direct communication.

(b) Relayed communication.

Fig. 1: Two practical paradigms of AFC communications.

In this study, we will show an approach of implementing relay attacks using commercial off-the-shelf (COTS) devices. We use a mobile phone to relay the communication of top-up transaction between the contactless smart card and the card reader and falsify the data to make the issuer believe that the top-up is failed. Demonstrated in Fig. 1, we use an NFC-enabled Android mobile phone as a relay proxy. The proxy then talks to a laptop via Wi-Fi to transmit and receive data that the terminal reads and writes. The actual data is reading from and writing to the card via an NFC card reader. As a result, not only we enlarge the communication distance between the terminal and the card, but we are also able to modify data during the communication.

We further conduct real-world attacks to the Beijing Munic-

978-1-5386-2784-6/17/$31.00 ©2017 IEEE

ipal Traffic Card (BMAC) system, *i.e.,* one of the most popular AFC systems. Specifically, a refund can be initiated after this attack. We have reported the attack to several popular AFC systems.

The major contributions of this study are as follow:

1) We analyze the weakness of ISO/IEC 14443-4 when facing a relay attack. The flaw appears quite general to all kinds of AFC systems following this standard globally.
2) We design a relay experimental method and perform the relay attack. The result shows that the protocol is vulnerable.
3) We propose two attack countermeasures, and discuss the feasibility and practicality of these countermeasures.

The rest of this paper is organized as follows. Section II reviews the related works. In Section III, we give the overview of how a top-up transaction is made. Section IV demonstrates the experimental methods for attacks. Section V discusses some countermeasures to this kind of attack and how transactions can be made more secure. Section VI presents our analysis about this kind of attack on AFC cards in other countries and how top-up can be made more robust even if attacks exist. And in Section VII we draw some conclusions.

II. Related Work

As we mentioned in Section I, researchers have been working on exploiting flaws in NFC. Haselsteiner and Beitfuß [9] showed a possible way to eavesdrop NFC. They suggested that, while normal communication distances for NFC are up to 10cm, eavesdropping is possible even if there is a distance of several meters between the attacker and the attacked devices. Extracting information from the transaction communication between a credit card and a POS terminal using eavesdropping is possible. However, this information (mainly credit card numbers, and expiration) can be obtained directly via NFC or even through social engineering. Paget [10] showed the process and later encode this information and write to magnetic stripe cards. This attack is also known as downgrade attack, which may not apply nowadays, due to banks have been working on refusing magnetic stripe cards and migrating to *Chip and PIN*. Other information transceived in the transaction communication is protected by secure keys. Eavesdropping in this situation is pointless.

The relay attack simply extends the communication distance between genuine terminals and devices. Relay attacks on NFC have been widely studied [11]–[13]. Initially, researchers built specific hardware to relay the communication between a smart card and a terminal. Hancke *et al.* [13] used a self-built hardware to enlarge the distance up to $50m$. They also deeply reviewed relay attacks in [11], discussing relay resistant mechanisms.

With the development of NFC, recent works have focused on relay attacks using mobile phones. Nokia 6131 was the first phone ever produced with NFC capability. Francis *et al.* [14] revealed the possibility to perform a relay attack using COTS devices. In [12], [14], [15], researchers performed relay attacks using Nokia mobile phones and discuss the feasibility of some countermeasures, such as timing, distance bounding, and GPS-based or network cell-based location.

More recently, researchers focused on relay attacks with Android mobile phones. Roland *et al.* [16] described relay attack equipment and procedures on Android phones. Dang *et al.* [17] described a scalable scenario for attackers to falsify AFC data.

III. Overview of a Top-up Transaction

As we discussed in Section I, since MIFARE Classic card was proved insecure, AFC cards have been being migrated to processor cards for security reasons. Among the processor cards, billions of cards in China have been issued, which makes it a very typical and good representative of AFC card system. In this section, we are focusing on processor cards in China. And we will discuss AFC cards in other countries in Section VI.

The most commonly adopted specification of the contactless smart card in China is named PBOC. A PBOC top-up transaction consists of two phases:

1) **Initialize for load**[1] in which the card is put in a state where it holds the transaction fare, and send a message authentication code (*MAC*) back to the POS terminal to ensure the integrity.
2) **Credit for load** in which the card verifies the *MAC* generated by the issuer, and the balance increases accordingly.

The involved principals are the card, the POS terminal, and the issuer. The whole process is illustrated in Fig. 2.

A. Secure Key System

Before getting deep into the two phases, we need to get known to the primary secure key system in PBOC. There are three master keys (MK) held by the issuer: master purchase key (MPK), master load key (MLK), and master TAC (transaction authorization cryptogram) key (MTK). Each card has an application serial number (ASN), which differs in different cards, for identifying a specific card. Using a key derivation function, each card holds its own derivated keys (DK) accordingly:

$$DK = 3\mathrm{DES}(ASN, MK) + 3\mathrm{DES}(\sim ASN, MK)$$

The purpose of keys in a card is to generate message authentication codes to verify data transceived during the transaction. Though keys differ in different cards, a derivated key will not be used directly to generate MAC in a transaction. Instead, a single engagement session key (SESK) is generated using transaction data to calculate MAC, which differs in each transaction:

$$SESK = 3\mathrm{DES}(data, DK) \quad (1)$$

The calculation of MAC is identical to EMV standard whose process is the same as ANSI X9.9 (ISO/IEC 9797-1).

All secure keys defined in PBOC are summarized in Table I.

[1]PBOC uses the term "load" for top-up.

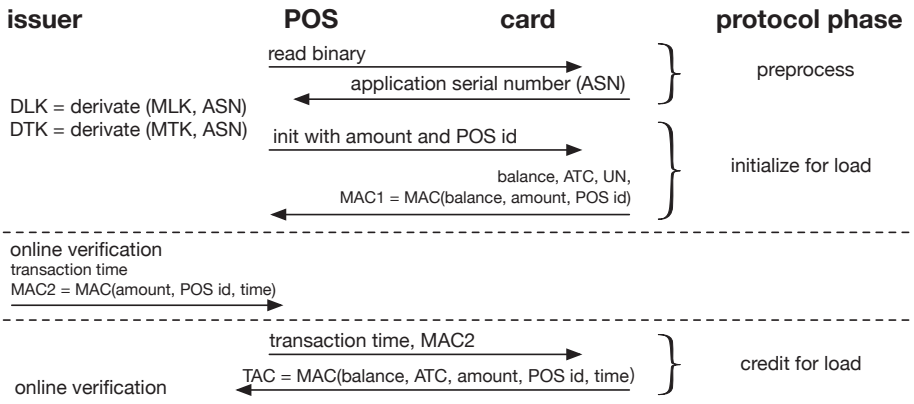


Fig. 2: Outline of a top-up transaction.

TABLE I: Secure key systems in PBOC

Key Usage	Issuer	Card	Session Key
Purchase	MPK	DPK (derived using ASN)	SESPK
Load	MLK	DLK (derived using ASN)	SESLK
TAC	MTK	DTK (derived using ASN)	-

B. Top-up Transactions

During the first phase, *initialize for load*, the POS terminal tells the card the amount of top-up and the POS terminal ID. The card preserves the amount, and responds with its balance, an application transaction counter (ATC - a 16-bit number stored in the card and increased in every transaction), a unpredicted number (UN), and a MAC. Among these data, the unpredicted number and ATC are used as input data of Eq. 1. The generated session key, known as SESLK, is applied to calculate the MAC, which is called MAC1.

Once these data is sent to the issuer through the POS terminal, the issuer verifies the amount and checks the integrity by recalculating the MAC1 over the received data fields. A valid MAC1 results in the following response: the transaction time and a MAC calculated from the amount, the POS terminal ID, and the transaction time.

The MAC2 authorizes the card to increase its balance, and generate a transaction record in the transaction history file. As a final step to confirm the success of a top-up, a 32-bit TAC is calculated and sent to the issuer immediately. The calculation is based on derivated TAC key (DTK), new balance, ATC, transaction amount, POS terminal ID, and transaction time. Once the issuer receives TAC, it is aware of the successful top-up.

IV. Attack Design And Implementation

In a successful top-up transaction, a genuine card generates a valid MAC1 as a proof. The card then relies on the issuer to provide a valid MAC2 to finish the transaction. Both communications are protected by session key SESLK through the message authentication code. Therefore, as a man-in-the-middle, it is impossible to modify any data as long as keeping master key secure.

However, we have discovered a major flaw that breaks the verification down. As we mentioned in Section III-B, a transaction authorization cryptogram (TAC) is crucial to issuer to confirm the top-up is successful. In a top-up transaction, a valid MAC2 will increase the balance of a card. However, *if and only if a valid TAC is provided*, the transaction succeeds. This design ensures no loss of the customer. Here brings the question: if somehow we are able to falsify the TAC, should the top-up be treated as a failure on the issuer side? More precisely, since the card has received a valid MAC2, the balance should increase as a consequence. But to the issuer, it fails. This sounds reasonable in theory, but is it viable in practice? We decided to figure it out.

A. Experimental Method and Results

Relay attacks against contactless smart cards have been discussed before, but there are many practical challenges for a real-world attack to work. In this section, we describe our approach: identifying an exploitable system, deploying a relay system, and performing the attack.

B. Performing the attack

We conducted a Moto X (XT1095) mobile phone to perform the attack. In our relay app, all commands received are directly sent to a laptop connected with a NFC card reader. The response is then sent from card reader through the laptop back to the app, and finally responds to the POS terminal. The whole process and equipment are demonstrated in Fig. 1(b).

We first performed a relay experiment using an app Alipay, which is the most popular electronic purse app in China, to top up our Yikatong card normally. The APDU (application protocol data unit) trace is listed and described in Table II.

The key steps in the trace are step 14 and step 15:

1) Step 14: initialize for load
   C-APDU: `805000020B00000003E8120080800001`
   - Amount: `000003E8`, 10.00 *yuan*
   - POS id: `120080800001`

   R-APDU: `00000B7C0005000057D8CC76392A6007`
   - Balance: `00000B7C`, 29.40 *yuan*

TABLE II: Command trace of a normal top-up

#	C-APDU	R-APDU	Explanation
1	00A40000021001	-	Select the EC file
2	805C000204	00000B7C	Read the EC balance (39.40 *yuan*)
3	00B2019C17	000794000B7C00011603041 10220080000100000000000	Read last top-up record
4	00A40000023F00	6F10840E315041592E5359532E4444463031	Select the master file
5	00B0840020	1000751132009883010200300000000 00000000000000002015101320211013	Read the card number (as ASN)
6	00B08C0801	01	Check if the card is forbidden
7	00B0850005	000000024E	Read the redundant transaction counter(590)
8	0084000004	0CE92186	Generate a random number for challenging
9	04D685000500000024FFA8FDB54	-	Update the redundant transaction counter
10	00A40000021001	-	Select the EC file
11	0084000004	776C244B	Generate a random number for challenging
12	04E200981B0000000003E80001151118 12008080000100000000000003A5279BB	-	Update top-up record
13	002000006313233343536	-	Verify PIN
14	805000020B00000003E8120080800001	00000B7C0005000057D8CC76392A6007	Initialize for load
15	805200000B2015111819104342FE26DC	0EB947B0	Credit for load

TABLE III: Error status code in load

Status Code	Explanation
6E00	CLA incorrect
6901	Command unacceptable
6985	Condition unsatisfied
9302	MAC invalid
9303	Application locked

- ATC: 0005
- Unpredicted number: 57D8CC76
- MAC1: 392A6007

2) Step 15: credit for load
C-APDU: 805200000B2016111819104342FE26DC
   - Transaction time: 20161118191043, 2016-11-18 19:10:43
   - MAC2: 42FE26DC

   R-APDU: 0EB947B0
   - TAC: 0EB947B0

In step 14, the terminal initialized a top-up transaction with 0x3E8 (1000 in decimal) as the amount. The minimum unit of CNY is 0.01 *yuan*, thus the amount is 10 *yuan*. The card then generated information together with MAC1 for authentication and integrity.

In step 15, the MAC2 returned from the issuer made the balance increased and as a result, a TAC with status code 9000 (we omitted in the trace due to code 9000 stands for success) was responded. According to the standard, the other response status codes are listed in Table III. As we mentioned in Section III-B, a successful top-up ends up with a correct MAC2 from the view of card but with a correct TAC from the view of issuer. What we expect is to increase the balance in the card but to make it failed in the issuer side. As a result, we decided to modify the response code of step 15 (credit for load) to 9302, indicating the MAC2 is incorrect.

Then we performed a relay attack using the same equipment. The top-up failed as we expected, shown in Fig. 3: the prompt on the screen means that the top-up has been interrupted and the user will get a refund.

Fig. 3: Relay attack leads a fail top-up.

C. Cause of the Problem

Now we take a deep look at the problem. In this attack, our relay equipment formed an unreliable link, in which we could modify data packets as we wished. Therefore there is no way to confirm a **common knowledge** that whether the top-up is finished successfully or not. This is quite similar to TCP handshaking, both of which deal with an unreliable link and have no way to confirm the top-up and the connection are successful or not.

In fact, this is a scenario of the famous two generals' problem, which is a thought experiment meant to illustrate the pitfalls and design challenges of attempting to coordinate an action by communicating over an unreliable link [18]. Two generals' problem is proved to be unsolvable [19]. As a result, it is impossible for anyone to design a protocol that works perfectly. However, we still have different countermeasures in different scenarios.

V. Defenses

Aforementioned two generals' problem in Section IV-C results in no solution to bypass the flaw in theory. However, we are still able to defend relay attacks in indirect ways. In this section we will describe two points for attention that should reduce the vulnerabilities.

No refund. During the relay attack in Section IV-B, once the issuer verifies MAC1, it will generates MAC2, which is

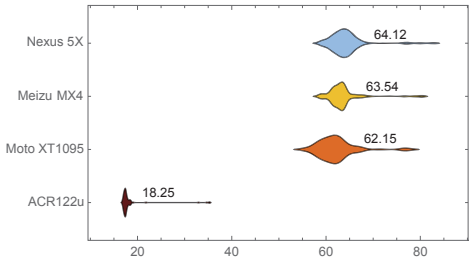


Fig. 4: RTT (in milliseconds) of different devices.

able to finish the top-up transaction in a genuine card. The TAC response from the card does not affect the top-up result. Though the second phase may fail, it happens occasionally. In fact, we tested the top-up transaction using 15 different mobile phones (including Moto, Samsung, Huawei, Google Nexus, and Meizu), and the result showed that among these phones, only Google Nexus 4 and Huawei G660 do not support this kind of transaction. The reason of failure is that the time interval of two phases is longer than the maximum timeout of two continuous commands, which is determined by the firmware. Now that it is uncommon to fail in phase 2, a mobile phone model blacklist can be used. Additionally, in order to avoid any possible loss, once the issuer generates a MAC2, there should not be automatic refund service provided. Instead, refund must be verified manually with a physical card provided, because the card, which is protected by several secure keys, itself shall not be forged.

Detect relay attack. As we demonstrate in Fig. 1(b), the relay process (additional wireless communication and HCE) introduces delay. Although the top-up via Alipay accepted the delay introduced by the attack, we wanted to quantify this delay in detail.

We select `GET RANDOM NUMBER` command and measure the round-trip times (RTTs) of them by: 1) using a ACR122u card reader directly, and 2) using different mobile phones as relay proxies. The result is shown in Fig. 4, which is collected from 100 tests.

The results show that it costs three or four times as transmission to a physical card. As a result, it is possible to detect relay attack by enforcing stricter timing restraints.

VI. Discussion

We have revealed the attack on AFC cards in China. In this section, we first discuss the possibility to attack AFC cards in other countries. Then we discuss other measures that the issuers can take to reduce the loss.

A. AFC cards worldwide

EZ-Link. The EZ-Link card is used for the payment of public transportation fares in Singapore. In October 2009, CEPAS, or Contactless e-Purse Application, was issued and deployed. The command/response in CEPAS also follows the convention in ISO/IEC 7816-4.

Top-up in CEPAS is simpler than PBOC, which requires a single `CREDIT` command. The 37 bytes data field in this command contains transaction amount, time, and other necessary authentication data. The response for a successful execution contains the purse balance, signed certificate, and counter data. The issuer is able to verify these data to confirm the top-up is successful. However, according to the specification, this command may also be responded with a failure status code. Consequently, the top-up state can be different to the issuer and the card if a relay attack exists.

Oyster. The Oyster card is the electronic ticketing used on public transport in Greater London. Since December 2009, all new Oyster cards use MIFARE DESFire EV1 chips. MIFARE DESFire is compatible with ISO/IEC 14443 Type A and ISO/IEC 7816-4. Besides, it also has its own command set for security, applications, and data commands. Similar to EZ-Link, MIFARE DESFire use a simple command `CREDIT`. Its data field is quite different; the value can be transferred in plain, enciphered, or MACed text depending on the communication mode. The response is a 1-byte status code, `0x00` for a successful operation, other for error.

Simply returning an error code will not work, due to `CREDIT` command is cumulated until a `COMMIT TRANSACTION` command is issued. As a result, we may falsify an error code when `COMMIT TRANSACTION` command is executed.

CIPURSE. CIPURSE is an open security standard for transit fare collection systems. This standard was established by the Open Standard for Public Transportation (OSPT) Alliance to address the needs of local and regional transit authorities for automatic fare collection systems based on smart card technologies and advanced security measures. CIPURSE has been deployed in several cities, including Barcelona in Spain, Perm in Russia, and Medellin in Columbia. The top-up in CIPURSE is similar to Oyster, which uses a single command `INCREASE_VALUE` to add value but requires a command `PERFORM_TRANSACTION` to finish the transaction. Again, the attack can be performed via a status code other than `9000` in the `PERFORM` phase.

Octopus. The Octopus card is a contactless stored value smart card for making electronic payments in online or offline systems in Hong Kong. Instead of ISO/IEC 14443, Octopus card uses the Sony FeliCa technology, which is standardized in ISO/IEC 18092. The FeliCa is totally different from ISO/IEC 14443 cards, which is more like a memory card and does not contain special commands for financial transactions. The only way to top up or purchase is writing a new balance directly. Therefore, Octopus does not provide any software or equipment for users to top up using a mobile phone or PC. Because of the usage of FeliCa, it is not possible to relay an Octopus card using Android mobile phones.

B. Further measures to reduce the loss

In Section V, we discussed countermeasures against this kind of relay attack. Based on current situations, attackers

might successfully get refund. In this section, we will propose further measures to reduce the loss once being attacked.

All kinds of card we mentioned before are protected by secure keys. Though the communication can be attacked, the card itself is still considered highly secure. Fortunately, the card may provide additional information for us to figure out the attack. We will discuss PBOC first.

When the top-up finishes, a PBOC card automatically increases its application transaction counter (ATC) and adds a transaction record containing a transaction counter (accumulated for both top-up and purchase transactions), the transaction amount, the transaction time, and the POS terminal ID. Therefore, once a next transaction without relay attack applied, in which the transaction history cannot be falsified, finishes, reconciliation with the history uploaded to the issuer will find out the proof of attack without any difficulty. However, the attacker may also falsify the transaction history using the same relay technique. There is a last line of defense: the session key SESLK used to calculate the MAC is generated using the ATC of each top-up, which is independent from the purchase ATC. Thus once the issuer detects a discontinuous ATC, which is always possible, it will be aware of the existence of the attack.

The same strategies can be applied to EZ-Link. The response of the `CREDIT` command is encrypted from a set of data including an independent add-value counter.

As a result, when online top-up service provided, it is necessary for the operator to verify the identity of the card holder, which makes it possible to track the attacker once the attack is detected.

VII. Conclusion

AFC systems are well-applied globally and billions cards are in issue. After MIFARE Classic card was cracked, processor cards are thought to be secure. However, the fact is not like that. In this paper, we discussed the practical relay attack on AFC cards in China, which may apply to other systems in other cities or countries. In a strictly standardized system, this attack is undefendable directly. But in realistic systems, we also provided countermeasures before and after the attack.

When ISO/IEC 7816-4 was created 20 years ago, it was originally designed for contact cards. Communication security is assumed because the card needs to contact with the reader directly. The compatibility of latter ISO/IEC 14443 reduced the cost for card migration. Unfortunately, this also introduced the potential risk of being attacked in various ways. The birth of NFC and HCE brought the convenience but also opened the door wide to attackers.

This flaw challenges current thinking about the security of near field payments. Despite that the loss caused by the attack described in this paper can be reduced by the countermeasures we gave, this relay technique can be used easily in other scenarios which may result in incredibly huge loss. It is time for the industry to take an interest.

Acknowledgment

This work is supported in part by the High-Tech Research and Development Program of China (“863 China Cloud” Major Program) under grant 2015AA01A201, the National Natural Science Foundation of China (NSFC) under grants 61471217, 61272429, 61432002 and 61632020, the CCF-Tencent Open Fund under grant RAGR20160105, and NSFC/RGC Joint Research Scheme under grant 61361166009.

References

[1] F. D. Garcia, G. de Koning Gans, R. Muijrers, P. van Rossum, R. Verdult, R. W. Schreur, and B. Jacobs, *Computer Security - ESORICS 2008: 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ch. Dismantling MIFARE Classic, pp. 97–114.

[2] G. de Koning Gans, J.-H. Hoepman, and F. D. Garcia, *Smart Card Research and Advanced Applications: 8th IFIP WG 8.8/11.2 International Conference, CARDIS 2008, London, UK, September 8-11, 2008. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, ch. A Practical Attack on the MIFARE Classic, pp. 267–282.

[3] N. Courtois, K. Nohl, and S. O’Neil, “Algebraic attacks on the crypto-1 stream cipher in mifare classic and oyster cards.” *IACR Cryptology ePrint Archive*, vol. 2008, p. 166, 2008.

[4] “Identification cards Integrated circuit cards Part 4: Organization, security and commands for interchange,” International Organization for Standardization, Geneva, Switzerland, ISO/IEC 7816-4:2005(E).

[5] “City union card of digital city General technology requirements,” Standardization Administration of the People’s Republic of China, Beijing, China, GB/T 31778-2015.

[6] “Specification for Contactless ePurse Application (CEPAS),” Singapore Standards Council, Singapore, SS 518:2014.

[7] “Contactless pre-paid/post pay IC card User card,” Korean Standards Association, Seoul, South Korea, KS X 6924:2009.

[8] “CIPURSE V2 - Operation and Interface Specification,” OSPT Alliance, Munich, Germany, CIPURSE 2.0.

[9] E. Haselsteiner and K. Breitfuß, “Security in near field communication (nfc). strengths and weaknesses,” in *In Workshop on RFID security*, 2006, pp. 12–14.

[10] K. Paget, “Credit card fraud – the contactless generation,” in *ShmooCon*, 2012. [Online]. Available: http://www.tombom.co.uk/Paget-shmoocon-credit-cards.pdf

[11] G. P. Hancke, K. E. Mayes, and K. Markantonakis, “Confidence in smart token proximity: Relay attacks revisited,” *Computers and Security*, vol. 28, no. 7, pp. 615–627, Oct. 2009.

[12] K. Markantonakis, “Practical relay attack on contactless transactions by using nfc mobile phones,” *Radio Frequency Identification System Security*, vol. 12, p. 21, 2012.

[13] G. P. Hancke, “A practical relay attack on iso 14443 proximity cards,” *Technical report, University of Cambridge Computer Laboratory*, vol. 59, pp. 382–385, 2005.

[14] L. Francis, G. Hancke, K. Mayes, and K. 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)*. Springer-Verlag, 2010, pp. 35–49.

[15] R. Verdult and F. Kooman, “Practical attacks on nfc enabled cell phones,” in *2011 3rd International Workshop on Near Field Communication (NFC)*, Feb 2011, pp. 77–82.

[16] M. Roland, J. Langer, and J. Scharinger, *Information Security and Privacy Research: 27th IFIP TC 11 Information Security and Privacy Conference, SEC 2012, Heraklion, Crete, Greece, June 4-6, 2012. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. Relay Attacks on Secure Element-Enabled Mobile Devices, pp. 1–12.

[17] F. Dang, P. Zhou, Z. Li, E. Zhai, A. Mohaisen, Q. Wen, and M. Li, “Large-scale invisible attack on afc systems with nfc-equipped smartphones,” in *2017 IEEE International Conference on Computer Communications (IEEE INFOCOM)*, April 2017.

[18] Wikipedia, “Two Generals’ Problem — Wikipedia, the free encyclopedia,” http://en.wikipedia.org/w/index.php?title=Two%20Generals’%20Problem&oldid=701668777, 2016, [Online; accessed on May 2, 2016].

[19] F. Kennard, *Thought Experiments: Popular Thought Experiments in Philosophy, Physics, Ethics, Computer Science & Mathematics*. Lulu.com, 2015.