How secure is it to pay with my smartcard wirelessly? - Analyzing the EMV standard

Benjamin Pfanz
Technical University Munich
Munich, Germany
benjamin.pfanz@tum.de

February 10, 2023

***Abstract*—The banking cards that are used for paying at shops or for withdrawing cash are called smartcards. While smartcards are also used as electronic passports or health cards, they play an important role in the payment sector.**

Almost all of the payment cards adhere to the EMV (Europay, MasterCard, Visa) specifications with more than 12 billion EMV smartcards in use as of 2022. The EMV standard is not a single protocol but it contains several variations. Some functionality is only described in a high-level view and the concrete implementation is up to the payment companies. As the EMV standard is quite complex defined on more than 2000 pages, this paper aims to give an understandable overview of the concepts. Especially the wireless case is discussed in more detail. The focus is on the security of the protocols including key and certificate management.

As EMV has great relevance to the global market and payment security, there is plenty of scientific work to analyze weaknesses. In general the EMV standard is comparatively secure in contrast to the previous magnetic stripe cards. However, due to the maximization of compatibility and different variants and implementations of the protocol, multiple attacks against EMV have been found. While other researchers usually discuss only the relevant part of EMV leading to their discovered weakness, this paper aims to provide a general understanding of security concepts in EMV and then reviews important attack vectors.

***Index Terms*—EMV, security, smartcard, contactless payment, vulnerability, certificates**

I. Introduction

Even though Germany is a country where cash is very popular, smartcards for payments are increasingly used. As opposed to other countries, bank customers in Germany are still cautious with card payments [1]. This may result from limited knowledge about the functionality of smartcards. Especially the wireless mode can cause worries as there is no need for physical interaction and the data is "somehow" transported over the air. This paper assesses the security of wireless payment and judges attacks that have actually been conducted. This should provide enough background knowledge to argue which fears may be justified and which are not.

Even if people use cash as one payment method, nobody disputes the influence of smartcards. Smartcard is a generic term not only for the payment cards or credit cards. Smartcards have been introduced in the health care system or as ID cards in many western countries. They are also used in public transportation or to access buildings. In this paper we only discuss smartcards used in the payment sector. While there are cards from different payment service providers, they almost all follow the EMV specifications. That is because EMV is used worldwide and 90% of all card-present chip transactions comply with the EMV standard (in Europe the adoption rate is even at 99%). Today more than 12 billion smartcards with EMV are in use globally [2]. Banks have a strong incentive to adopt to EMV as a *liability shift* has been introduced. Cards following the EMV standard are considered secure according to this shift and customers can not get a refund of transactions from the bank even if the card was stolen. The argumentation is that a card is useless unless the attacker knows the PIN. Some researchers question this liability shift as they found vulnerabilities that the customer cannot do anything about [3], [4]. In examining smartcard payments we will therefore especially analyze the EMV protocol.

very interesting!

II. History

Before the introduction of smart cards with a chip in the form that is common today, magnetic stripe cards dominated the market. Magnetic stripe cards are able to store several bytes of information. This data can be written or read by specific devices and is encoded magnetically. The problem form a cryptographic perspective is the fact that a magnetic stripe can only store data and is not able to compute anything. This means that each reader capable of reading these standards is able to read the entire data data stored on the card. An attacker can easily read out all the information and clone a second card with it. This cloning attack was a major vulnerability within the design of magnetic stripe cards itself [5]. Another problem is enabling a method for checking the entered PIN. As the magnetic stripe card is not able to compute anything it has to reveal the PIN to the terminal to verify it [6]. This is insecure as the PIN may be revealed to a not authorized entity. There have been countermeasures introduced to reduce the fraud done with these cards: By analyzing the transaction databases, banks can be alarmed when anomalies happen. The payment network for example could issue an alarm when the card is used in a foreign country. Some measures have also been proposed to counteract cloning of magnetic stripe cards. By storing fingerprint data instead of a PIN it would be harder for attackers to use a cloned card [7]. However, these countermeasures do not solve the original problem: the magnetic stripe still has all information just stored on the card and does not perform computations.

The tremendous advantage of smart cards now is the integrated chip: It has the ability to execute computations just like a normal processor does - the computational power is lower though. PINs for example can now be checked within the card itself without revealing it to the terminal [8].

III. Smartcards and EMV

Before analyzing the protocol structure of an EMV transaction in detail, this chapter presents an overview over the whole payment system: What technology is used in the smartcards and how is the payment infrastructure set up? The next chapters introduce then the organization EMVCo behind the EMV, certificates in the EMV infrastructure, a detailed explanation of the protocol and types of attacks and their impact.

A. Structure of a smartcard

A smartcard is called “smart” because it is capable of performing computations with the built-in chip compared to the previous magnetic stripe cards which were only able to store data. This chip needs an interface for transmitting data and requires electricity for functioning. Most smartcards in the payment sector support two different physical modes: In the contact mode the electrical contacts on the chip are connected to the terminal and provide electricity to the chip as well as information. The contacts have different roles like power supply, ground, clock or transmitting bit sequences as an I/O interface [8]. In the contactless mode the card receives power from electrical induction. Thin wires around the borders of the card form an electrical coil. If this coil is within a changing magnetic field, electricity is introduced in the wires and powers the chip. Modulations of this electromagnetic field can transmit information. Smartcard and terminal use the Near Field Communication (NFC) standard to exchange data this way. NFC is closely related to Radio Frequency Identification (RFID) where the latter is mainly used to identify and track goods by using RFID tags attached to these. RFID systems work in different frequency ranges whereas NFC can be seen from a technical level as the RFID standard at 13.56 MHz. While RFID tags can transmit data over various distances based on the used frequency, NFC devices are able to communicate only up to 10 cm. On the application level however, NFC is more similar to Bluetooth. RFID uses a concept where active readers read out data from passive transponders; the information flow is unidirectional. NFC builds on top of RFID technology but allows more sophisticated data transmissions in both directions [9].

B. Overview of an EMV payment

While the communication between card and terminal happens via contacts or the NFC protocol locally, there are more actors involved in an EMV payment: Figure 1 depicts the participating entities of an EMV transaction [10]. In the following the term Smartcard is often abbreviated as Card. This is the card a customer receives from a bank after opening a bank account. There is not only one type of card, though. A major difference is between debit and credit cards. The first type only allows money to be transferred if it is already on the customer’s bank account while the second one also allows a certain amount of debt [11]. We will not elaborate on such distinctions of smartcards as they do not affect the general message flow during a transaction.

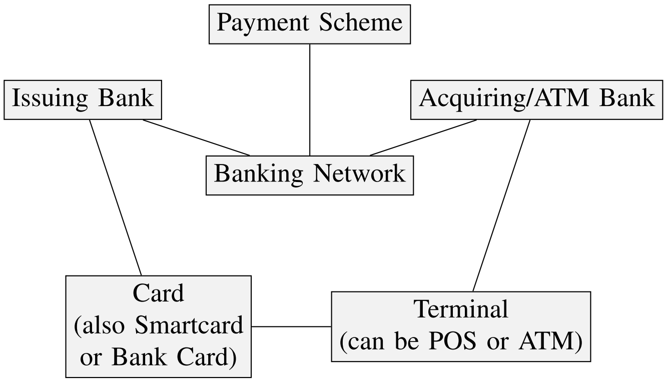

Fig. 1. Links between the involved actors in an EMV payment system.

The second actor in every EMV transaction is the terminal. This can be a Point Of Sale (POS) terminal which is typical at e.g. supermarket checkouts. An Automated Teller Machine (ATM) works according to the same principle. You can withdraw or deposit money at an ATM. We will refer to these in the further discussion generally as a terminal as it does not matter in terms of EMV protocol flow.

The actual money transfer is executed in the banking network between the issuing and the acquiring bank. The messages between the banks follow a specific payment scheme which is a set of rules the actors in the banking network have to agree on [10]. We will - however - put our focus on the interaction between card and terminal as most of the scientific literature reports frauds and vulnerabilities there and not in the banking network.

C. High level view of the EMV standard

The EMV standard builds on top of other standards depending on the physical mode of the smartcard: The International Organisation for Standardisation (ISO) defines the ISO/IEC 7816 for Integrated Circuit Cards (Smart cards) and the ISO/IEC 14443 for Contactless Integrated Circuit Cards. EMV follows these specifications and exchanges payment specific messages on top of those two standards [12], [13].

EMV is not a single protocol but rather a toolbox with predefined parts for building EMV compatible protocols. Following all the instructions of the bare EMV standard does not yield in a working protocol. There are multiple possible configurations to choose between and some functionality is even to be implemented by every company on their own. This results in some of the code - especially specific implementations - being proprietary.

The core standards are usually publicly available and covered in multiple books available on the EMVCo download

Webpages. The contact specifications are described in four books [14]–[17]. General contactless specifications can be found in [18]–[20]. Additionally there are 8 separate books for kernel specifications [21]–[28]. These describe some functionalities for the contactless EMV protocol but a card does not have to implement all of them but usually only one. The different brands MasterCard, Visa and other payment service providers have each released their own contactless kernel specifications.

Besides the EMV contact and EMV contactless specifications, there are several other protocol blocks a part of EMV: An example is EMV-CAP, a standard for using EMV smart cards to carry out transactions in online banking [29]. However, we want to concentrate on the EMV contactless specifications in this paper.

IV. Behind EMV: EMVCo

A. EMVCo organization structure

The EMV standard is specified, managed and actively developed by a limited liability company called "EMVCo". Originally this organization was founded in 1999 by the three companies Europay, MaterCard and Visa - hence the name. Between 2004 and 2013 four more companies, namely JCB, American Express, UnionPay and Discover joined the organization. Because Europay has been acquired by MasterCard, there are six companies in 2022, each of which holds an equal share in the EMVCo LLC [12], [30].

Within EMVCo, operational tasks are coordinated by the Board of Managers, the Executive Committee on the other hand provides new strategic focus. These groups consist of representatives from the six member companies. Third parties are also able to monitor or to participate in the decision making process [12], [31]:

- EMVCo Subscriber can be any party that wants to monitor the current development process. They have also the opportunity to provide feedback. Currently there are 426 Subscribers registered [32].
- EMVCo Associates are organizations contributing expertise for new EMV Specifications. Among those Associates are big tech companies like Google, Microsoft or Amazon as well as several banks or payment organizations like the Bank of America, globalpayments or the Dutch Payments Association [32].
- The Board of Advisors is the most active form of engagement: Participants here advice EMVCo on strategic questions and have a voting right on specification publications.

As an entity defining new standards, EMVCo also works together with other standards organizations:

- The Payment Card Industry Security Standards Council (PCI SSC) is called "complementary in enhancing payment security". However it is not very clear why a second organization is required given that PCI SSC is founded by American Express, Discover, JCB International, MasterCard and Visa Inc. The members and organization structure is very similar to EMVCo but obviously PCI SSC concentrates especially on the security of payment systems [33].
- EMVCo collaborates with the Near Field Communication (NFC) Forum to develop EMV specifications for mobile devices in addition to smartcards. As our focus is the contactless mode of smartcards and not payment with mobile phones, we do not go into detail here [12].

B. The mission of EMVCo

The goal of EMVCo is "to facilitate a payments infrastructure that is standardised in terms of security and interoperability" [34]. That means, EMV should enable secure and seamless transactions worldwide. To achieve this, the organization develops and manages EMV Specifications. However there are neither enforcements for companies to comply with these standards nor does EMVCo develop products themselves such as smart cards or terminals [12], [34]. Instead they just provide specifications on certain behaviors or data elements that are sufficient to carry out an EMV transaction. These define how a card and a terminal application must behave to complete a transaction. Furthermore they set implementation options but do not present one complete and ready implementation of the protocol [12]. Individual payment systems are required to build their own concrete specifications on top of the EMV standard which can include some proprietary formats specific to each company [12]. A smartcard must therefore support some functionalities defined by EMV but can execute additional internal computations that are not mentioned in EMV.

V. Certificates in EMV infrastructure

Before analyzing an EMV transaction and the typical involved steps, this section aims to provide an overview of the key management and the certificate infrastructure in this chapter. While other scientific papers focus on specific vulnerabilities and explain only details relevant to those, this section provides a general understanding of the cryptography that is used in the EMV setting: To certify all the involved entities (cards, terminals, banks), there exists an own infrastructure for EMV. Public key infrastructures (PKIs) are common in cryptography for networks today. The probably most common standard uses X.509 certificates for ensuring secure internet communication [35]. EMV does not interact with this standard but has its independent PKI.

A PKI network is strictly hierarchical with the Certificate Authority (CA) as the highest instance as depicted in Figure 2. This is the trust anchor in the network which means that every party trusts the CA and what it signs or validates. EMV supports multiple CAs that can be either compliant with Visa or MasterCard. The terminal stores the Certificate Authorities locally just as a browser stores the trusted CAs [36].

A. Authentication of the issuing bank

A bank that issues EMV cards is certified by one of the CAs: In the certification procedure, the issuing bank first sends a certificate-request to the CA. Then it receives a so-called

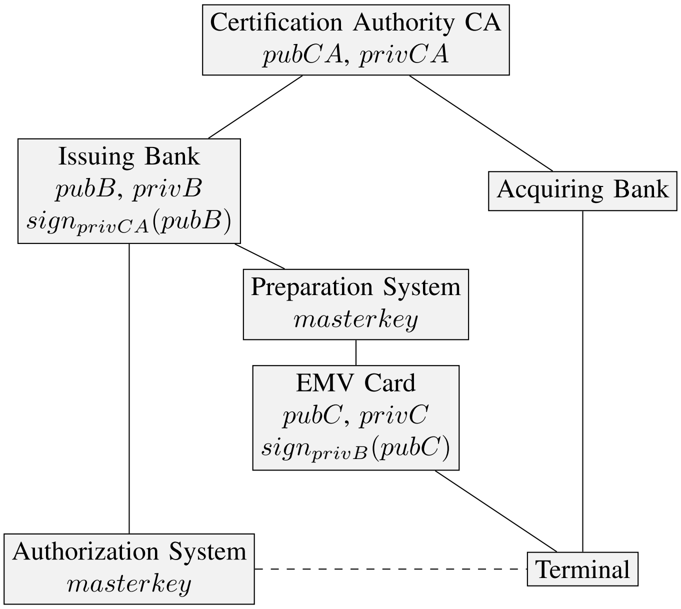

Fig. 2. EMV PKI infrastructure

issuer certificate. This is a document containing the public key of the bank signed with the private key of the CA. Every entity trusting the CA can therefore also trust the PK of the bank which is officially signed by the CA. To verify the bank's certificate, the terminal can decrypt the certificate with the publicly available $pubCA$. For generation of the asymmetric key pair, the common RSA algorithm is used [37].

B. Authentication of the card

With a verified card issuer, the card can also be authenticated. There are three methods for the authentication of the card to the terminal [3], [29]:

- Static Data Authentication (SDA): the card provides the Signed Static Authentication Data (SSAD) to the terminal. This is a signature over the static data of the card, such as the static primary account number (PAN). The signature has been computed during production of the card with the private key of the bank $privB$. The terminal can verify that the static data of the card has not been altered but is indeed from the issuing bank. An attacker can not modify data this way but could clone the card (or execute a replay attack) as they can just copy the signature without knowing the private key of the bank.
- Dynamic Data Authentication (DDA): The dynamic part is here a random number from the terminal, the so-called Unpredictable Number (UN). For this method, the smart card must have its unique asymmetric key pair as the signature of this random number is not possible in advance. Similar to the procedure between the issuing bank and the CA, the card can also get a certificate - now from the (verified) bank. For this certificate, the public key of the card $pubC$ is encrypted with $privB$ of the bank. The card now signs the UN together with an own random number (NC) with its private key $privC$ and sends the Signed Dynamic Authentication Data (SDAD) back to the terminal. The terminal can ensure that the card has the private key $privC$ by decrypting the SDAD with $pubC$. By verifying the signature it can also ensure that the key pair is genuine and not created from an attacker. This method prevents cloning because a malicious entity would need the private key of the card which is never transmitted.
- Combined Dynamic Data Authentication (CDA): Both of the above methods have the problem that they do only authenticate the card in a first step. The transaction however is carried out afterwards and the card decision if the transaction should be approved or denied is not itself verified. A possible attacker in a MitM position could forward all messages related to SDA or DDA to the genuine card and then - after the card is authenticated - e.g. wrongly deny a transaction. CDA improves this by including the card decision into the signature of the above DDA variant. An attacker can not alter any transaction decision as this is now included in the signature.

C. Additional keys

The three levels of authentication are not mandatory to be implemented in a smart card. The only minimum requirement is a 3DES symmetric key between card and bank (though virtually all cards today support the three authentication methods above). This key is used for the so-called Application Cryptogram (AC). The AC is a message encrypting several transaction details and is sent to the banking network at the end of a transaction. The bank can then verify if the card actually wants to transmit the money (because the message is encrpyted with the symmetric key only bank and card share). The key is also used to encrypt a response from the issuer if the card and terminal request an online authorization via the bank [37].

There are two additional symmetric keys a card issuer can implement into the card: One key for encrypting scripts and one key for the generation of a MAC in this procedure. Some transactions are carried out offline but in the other cases there is an online connection. During the authorization of a transaction, the card issuer can execute scripts on the card or transmit an update. This is encrypted with the symmetric key for scripts and the integrity of the script is checked with a MAC (for which an additional symmetric key is used).

The infrastructure for distributing and checking these symmetric keys can be chosen by the issuing bank. Often the bank uses a dedicated authorization system for validating the keys. For the production of the cards a preparation system is used to generate the symmetric keys. Both systems have to communicate the keys before the usage of the card. To retrieve unique symmetric keys, they are often derived from a master key for example using the unique account number. But this procedure is not part of EMV - every bank that issues such cards can design their own way of interaction between cards and bank and if they want to update cards or execute scripts [37].

D. Authentication of the terminal

Surprisingly the terminal is not authenticated in EMV transactions [29]. While for example electronic passports or electronic driving licences do enforce this, it is not required for EMV terminals. Indeed the authors of [29] view this as a major core flaw in the protocol design of EMV. By not authenticating the terminal it is also not possible to establish a secure channel. This opens the way for several attacks because they exploit the fact that not all messages are authenticated. This would change with the usage of a secure channel.

For terminals the following procedure is applied: Every vendor of EMV compliant terminals has to certify their products, i.e. the single terminal versions. There is a Level 1 and 2 testing to ensure compliant mechanical and electrical properties as well as correct software functions [38]. An additional Level 3 testing makes sure that the terminal can communicate correctly with the banking system [39]. This procedure does not ensure that a terminal manufacturer puts malicious parts into a terminal after a certificate was issued for the benign version. But as all manufacturers are registered with the organization EMVCo, it would be a high risk for a company to cheat as that may ruin their business.

A similar step applies to merchants. While it is possible nowadays to set up a Point of Sale system with just a smartphone or tablet and use it as terminal, the merchant has to register in every case. Registering information typically includes the contact details of the owner and the location of the shop. With this information and with the fact that every transaction is recorded in the banking network, police can quickly track down the responsible merchant in case of cheating.

Defrauding the customer as a terminal owner is generally possible. One attack vector would be a compromised terminal that shows a lower amount on the display than is communicated with the card. But with the registration of both terminal manufacturers and merchants these attack types are not attractive. why does owner have to register?

Therefore the authentication of a terminal would make more sense from a security point of view but the process also works without it.

VI. A POSSIBLE EMV PROTOCOL PROCEDURE

This chapter provides an in-depth view of a typical EMV transaction. Unless other scientific work, this paper tries to not only provide the exchanged commands and messages but also gives understandable descriptions of the steps depicted in Figure 3. As there is not a single EMV transaction data flow, we will discuss some alternative variants. Especially the Visa contactless protocol is mentioned sometimes as there is a major protocol flaw we will see in subsection VII-A.

An EMV transaction can have the following elements [3], [4], [29]:

1) An EMV transaction starts with the application selection. In the contactless mode, the terminal sends the `SELECT` command, followed by the string 2PAY.SYS.DDF01 (in bytes) to the card. The terminal sends a slightly different byte string if the transaction is carried out in the contact mode.
2) The card then returns a sequence of Application Identifiers (AIDs). One card can have multiple applications to choose from (e.g. Credit card or Debit card application, ATM mode to interact with an Automated Teller Machine) [3]. The AID is not part of EMV itself but it is a feature of the underlying ISO standard (ISO/IEC 7816). Other protocols like SIM card applications do also use (other) AIDs. There is not a single EMV application though, but each product issuer has multiple own applications. The AID consists of a five byte long registered application provider identifier (RID) followed by a proprietary application identifier extension (PIX). Every application provider gets their individual RID number from a registration authority defined in [40]. They can then assign different PIX values for different applications they have. Method Confusion!
3) The terminal selects one of the offered applications. In the case of a Visa transaction this AID number starts with A000000003 (as the RID for Visa is this number). Depending on the concrete Visa application, the following bytes (PIX) can differ.
4) The card asks for the Processing Data Object List (PDOL). This is a list containing all relevant transaction data the terminal knows: This includes transaction amount, country code, currency, date, transaction type and the Unpredictable Number (UN). The UN is a random number which the terminal generates to perform Dynamic Data Authentication and is optional (see subsection V-B).
5) The terminal sends the PDOL data if it is requested and issues `GET PROCESSING OPTIONS` to find out what processing options (especially Cardholder Verification Methods, see step 9) the card supports.
6) The card then sends the Application Interchange Profile (AIP) where the supported authentication and Cardholder Verification Methods (CVMs) are specified. There are however variations of EMV where e.g. the CVM is not written in the AIP but in the unauthenticated Card Transaction Qualifiers (CTQ) like in the contactless Visa mode. Additionally, the Application File Locator (AFL) is sent to the terminal. It shows which files from the file system of the card can be read. The card also sends proprietary Issuer Application Data (IAD) that is sent to the bank in case of an online transaction. The card can send an Application Cryptogram (AC) here which indicates that the card approves or declines a transaction already in this step.
7) The terminal sends the `READ RECORD` command and reads the smartcard files where the AFL points to. This includes information like the unique Primary Account

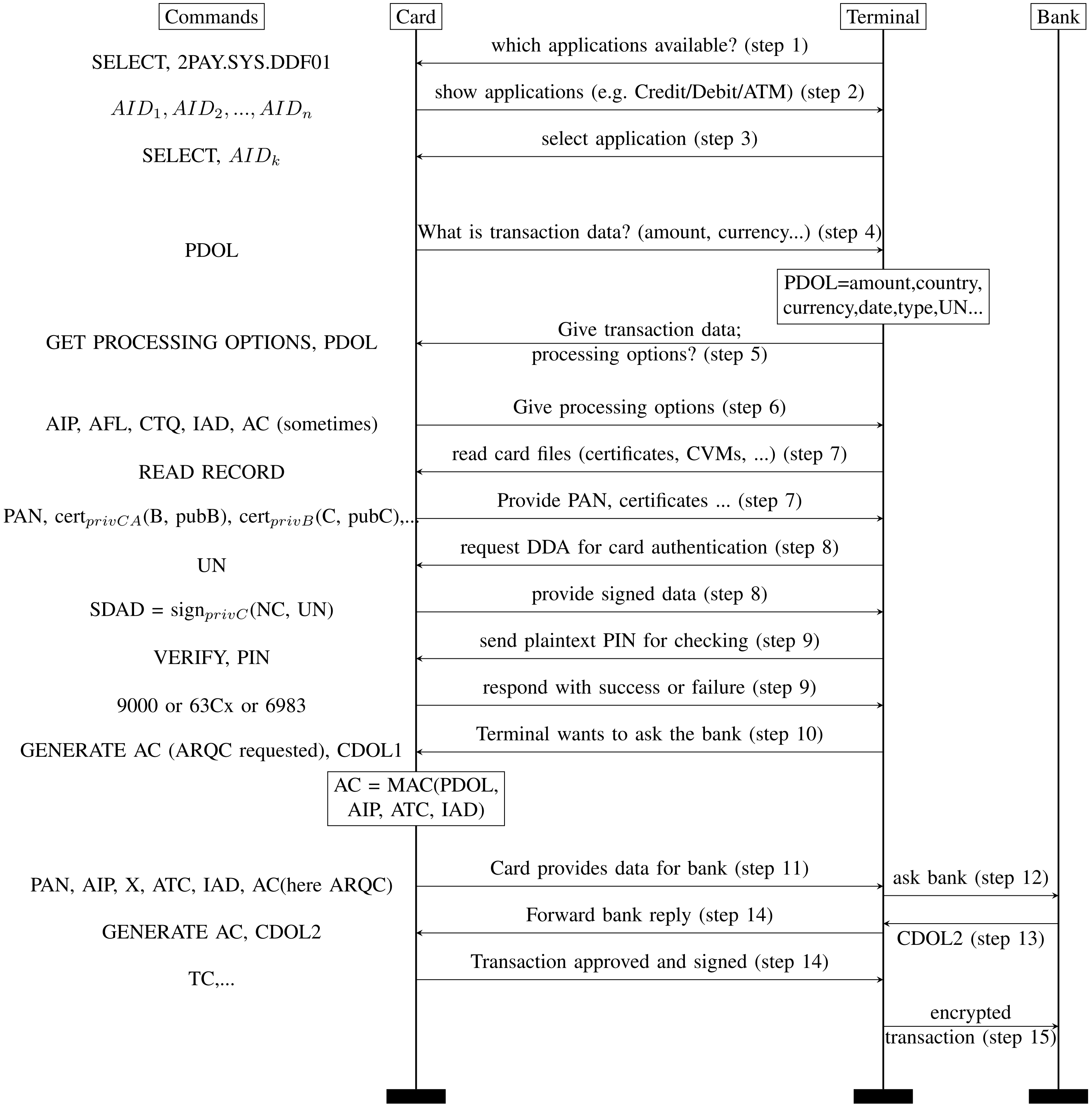
Commands
Card
Terminal
Bank
which applications available? (step 1)
SELECT, 2PAY.SYS.DDF01
show applications (e.g. Credit/Debit/ATM) (step 2)
$AID_1, AID_2, ..., AID_n$
select application (step 3)
SELECT, AID_k
What is transaction data? (amount, currency...) (step 4)
PDOL
PDOL=amount,country, currency,date,type,UN...
Give transaction data; processing options? (step 5)
GET PROCESSING OPTIONS, PDOL
Give processing options (step 6)
AIP, AFL, CTQ, IAD, AC (sometimes)
read card files (certificates, CVMs, ...) (step 7)
READ RECORD
Provide PAN, certificates ... (step 7)
PAN, cert$_{privCA}$(B, pubB), cert$_{privB}$(C, pubC),...
request DDA for card authentication (step 8)
UN
provide signed data (step 8)
SDAD = sign$_{privC}$(NC, UN)
send plaintext PIN for checking (step 9)
VERIFY, PIN
respond with success or failure (step 9)
9000 or 63Cx or 6983
Terminal wants to ask the bank (step 10)
GENERATE AC (ARQC requested), CDOL1
AC = MAC(PDOL, AIP, ATC, IAD)
Card provides data for bank (step 11)
PAN, AIP, X, ATC, IAD, AC(here ARQC)
ask bank (step 12)
Forward bank reply (step 14)
GENERATE AC, CDOL2
CDOL2 (step 13)
Transaction approved and signed (step 14)
TC,...
encrypted transaction (step 15)

Fig. 3. Example procedure of an EMV transaction. As there are multiple variants of EMV the protocol flow can vary. This scheme shows the DDA method for Offline Data Authentication (step 8) where instead SDA or CDA could performed. As CVM method the Offline Plaintext PIN is shown. According to step 9 there are other possibilities for CVM including online PIN where an interaction with the bank would take place. For transaction authorization the diagram depicts the variant where the terminal requests online authorization (step 12).

Number (PAN) or the card's expiration date, other static data and a list of supported CVMs. Further all certification data is transmitted here: the index of the Certificate Authority (CA), the public key of the bank and of the card (with their respective certificates). With the index of the CA the terminal can retrieve the public key of the CA from a local data base and verify the certificate of the bank. It then uses the public key of the bank to verify the certificate of the card. This process is elaborated in section V.

8) In the next step the card is authenticated by Dynamic Data Authentication (DDA) which includes signing the Unpredictable Number (UN) from the terminal and a random number from the card itself (NC). There are also other methods possible at this stage (namely Static Data Authentication, SDA, or Combined Data Authentication, CDA) which are described in detail in section V.

9) Now the card is authenticated and trusted. As a next step the person using the card must be verified as the genuine owner. For this purpose, one of the Cardholder Verification Methods (CVMs) is applied. Here it becomes very clear that compatibility is key for the EMV protocol. There are multiple possible CVMs:

   Paper signature: The merchant must check a handwritten signature in this case and verify the owner. However one can learn a foreign signature quite easily. This method is rarely offered today as the merchant is liable for any fraud by applying this verification.

   PIN: The PIN method is very common, but there exist three different ways how the PIN can be verified. With Offline Plaintext PIN the terminal sends the PIN in plaintext to the card and waits for a success message. The card can send the code 9000 for success, the code 63Cx for failure (where x={3,2,1} stands for the remaining tries) or the code 6983 in case the PIN is blocked as the maximum number of PINs has been entered. With Offline Enciphered PIN the terminal first requests a random number from the card. The PIN is now added to the random number and encrypted with the public key of the card. The third method is Online Enciphered PIN, where the terminal directly sends the encrypted PIN to the bank. The card is not involved in this version.

   The Consumer Device CVM (CDCVM) can not be used for smartcards but only for mobile phone payments, where the owner is verified directly by the device. The phone usually verifies the owner by fingerprint, face identification or a PIN entered into the smartphone. After successful verification the phone responds with a success message to the terminal.

10) Transaction Authorization is the last step. The terminal asks for an Application Cryptogram (AC) of the card. The terminal can request three different cryptograms:

   - An Application Authentication Cryptogram (AAC). This is the case when the terminal rejects the transaction. The card can do nothing about it but has to send the AAC (or theoretically abort the connection).
   - A Transaction Cryptogram (TC) in the case the terminal is ready to accept the transaction.
   - An Authorization Request Cryptogram (ARQC) if the terminal wants to ask the bank (and thus proceed with an online authorization).

11) The card can deviate from the desired cryptogram if it does not want the transaction to be carried out offline. However, the card can only send a cryptogram for a more secure procedure. An AAC or ARQC as an answer to a TC request is possible as the terminal would have accepted the transaction anyway. The card assesses the risk of the transaction with the provided CDOL1 data (Card risk management Data Object List) which includes transaction details. The cryptogram that is then sent by the card is a Message Authentication Code (MAC) over transaction details and additional information with a symmetric key (the AC key, see section V) that only the card and the bank know.

12) If the card has sent the ARQC as the Application Cryptogram, then the transaction must be authorized online. For this, the ARQC together with other transaction data (PAN, AIP, PDOL, CDOL1, ATC, IAD, AC and if necessary the encrypted entered PIN) is sent to the bank. (Note that this data must be forwarded by the terminal as the card and bank are not able to communicate directly. The terminal, however, can not change messages in this step as the ARQC protects the integrity of the transmitted data with the symmetric key between card and bank.)

13) The bank sends back an Authorization Response Code (ARC) that either allows or prohibits the transaction. It adds an Authorization Response Cryptogram (ARPC) to make the ARC answer tamper resistant. The ARPC is a MAC where the key is derived from the increasing Application Transaction Counter (which only bank and card know). The data from the bank forms the CDOL2 which the terminal forwards to the card.

14) The card now calculates a second Application Cryptogram and issues either an AAC if the bank declines or an TC if the bank accepts the transaction.

15) After a successful transaction the terminal forwards the Transaction Cryptogram to the bank and this cryptogram is a guarantee for the merchant that the bank has to execute the transaction.

VII. Types of attacks

After we have presented the security concepts and the protocol flow of EMV in detail, this section classifies some of the attacks that have been executed on real or test infrastructure. The attacks are grouped by different types. Most of the exploited mechanisms are general issues and are not specific to EMV. They often arise because the protocol flow of EMV allows many deviations and variants. This paper focuses on two main attack vectors that have been conducted in the past against payment systems - Man in the Middle and relay attacks. Several more attack vectors have been found including downgrade, pre-play or eavesdropping attacks. The papers[13] or [10] provide additional information.

A. Man in the Middle MitM

A Man in the Middle (MitM) attack is not specific to EMV. It is possible whenever an attacker can put themselves in a position where they can control the data flow between two parties. A MitM attack is not only possible in the contactless mode but there it is much easier to place an actor between terminal and card.

There is a widely recognized paper [4] where a MitM attack was successfully executed: The researchers could finish a transaction with entering any arbitrary PIN.

good example!

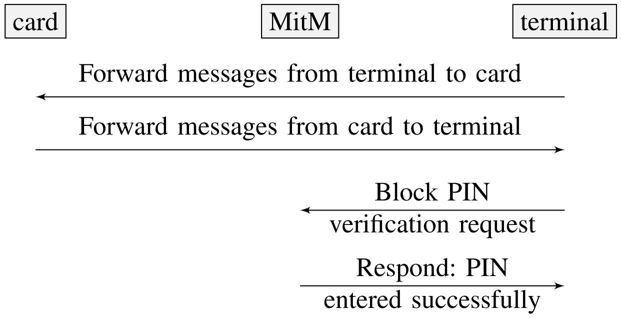
card
MitM
terminal
Forward messages from terminal to card
Forward messages from card to terminal
Block PIN verification request
Respond: PIN entered successfully

Fig. 4. Man in the Middle attack for allowing arbitrary PINs: Attacker forwards all messages, only PIN verification request is not forwarded to be checked by the real card but MitM sends directly success message. While this attack is not possible any more, there is a newer version for Visa cards in the wireless mode where the Attacker response is a successful Consumer Device CVM.

The problem here lies in the PIN verification step (compare Figure 4). For the authentication of the card itself in a previous step the attacker can simply forward all messages between terminal and genuine card. The PIN is then blocked from reaching the card. The vulnerability is possible because the EMV protocol allows different cardholder verification methods (CVMs). So the card thinks that PIN was either not needed for the transaction or the terminal does not support it and e.g. paper signature has been performed. The MitM device now sends a response to the terminal that the PIN has been entered correctly. When the paper was written, the card response to the entered PIN was transmitted without authentication. The terminal is not able to detect if the answer is from the genuine card or from a malicious device as it is not authenticated. This applied only to offline PIN verification as in online PIN verification the encrypted PIN is directly sent to the issuing bank and checked there [4].

In the paper the attack was only discussed for the EMV contact case - the wireless mode was not relevant in 2010. A real world attack was thought to be unfeasible due to the difficulty of miniaturization of a MitM in the contact mode. The attack was executed with additional hardware for demonstrating purposes of the paper. The card was wired visibly to a notebook and would have therefore been noticed in an attempted fraud. However, in 2011 a forensic analysis of stolen cards revealed that this vulnerability has actually been exploited [41]. The chip of a genuine, stolen card was soldered top to bottom with a second chip. The card was slightly thicker than a normal one but could be plugged into a terminal. The second chip was a free programmable one which is normally used in hobby electronics and utilized here as a MitM. The MitM chip communicates directly with the terminal and forwards all requests to the genuine chip except for the PIN verification request. In this case, there is always a success message sent. The authors call it the "most sophisticated smart card fraud encountered to date" [41]. Around 600000 € were stolen with 40 manufactured cards. After revealing the mechanism based on forensic analysis of confiscated cards, countermeasures could be introduced very quickly. The malicious chip verified every PIN even if there was no transaction going on. A software update was deployed to the terminals which recognizes cards that wrongly answer out-of-context PIN requests.

Method Confusion

While the specific attack with the soldered chip does not work any more, the original vulnerability can still be exploited in 2022 as verified in [3]. The attack is only possible for cards that support neither asymmetric cryptography nor online PIN verification, though. Most current cards have implemented at least one of those but the authors of [3] found another MitM vulnerability even for some of the newest cards. Their attack works in the contactless mode. As the payment providers have different protocols specified in their contactless mode, this attack does not apply to all cards. Only cards following the Visa, Discover or UnionPay kernel are affected while the last two have not been tested. There the Card Transaction Qualifiers (CTQ, see step 6) are not authenticated. The card tells the terminal in the CTQ which Cardholder Verification Method (CVM) should be used. For other EMV variants the CVMs are either included in the authenticated Application Interchange Profile (AIP) message or the CTQ itself is authenticated. The MitM now pretends to the terminal that the card wants to use CDCVM (Consumer Device Cardholder Verification Method). This method has been introduced to enable the verification directly on the smartphone (e.g. via fingerprint) for smartphone NFC payments. The attack enables a malicious person to carry out a transaction of high value where a PIN would normally be required. The authors of [3] use two mobile phones as MitM. One phone is held near the victim card and the other phone is used at the terminal as if the customer would pay via smartphone. Messages between the phones are transmitted over WiFi. This attack does not work for MasterCard transactions as there the CVM is not transmitted in an unauthenticated CTQ. Instead, it is written in the AIP which is authenticated.

B. Relay attacks

For contactless payment there is another type of attacks which occurs in different variants: Relay attacks.

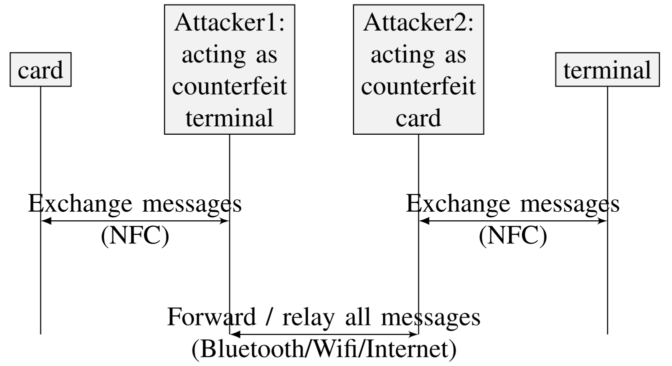
card
Attacker1: acting as counterfeit terminal
Attacker2: acting as counterfeit card
terminal
Exchange messages (NFC)
Exchange messages (NFC)
Forward / relay all messages (Bluetooth/Wifi/Internet)

Fig. 5. Scheme of a relay attack: In the simplest setting the two attacker devices forward (relay) all messages to make the real terminal and card think they are communicating with each other (although they may be far away when the attackers forward the messages via the internet.

Similar to a MitM attack, the attacker sits between the terminal and the card. While an MitM attacker is able to read or modify messages, this is usually not done in a relay attack. Instead, the attacker simply forwards all messages between the two sides [42]. What makes a relay attack powerful is that the attacker can extend the range and make two devices communicate which are very far away. While contactless payment with a card is only possible within 10cm from the terminal, the attacker can overcome this restriction by using two devices: One device acts as a counterfeit terminal and one as a counterfeit card (see Figure 5). Both counterfeit devices can be mobile phones with corresponding software. The attack is then executed by holding one of the mobile phones near a genuine card and the other mobile phone to the NFC reader of a terminal. The real devices exchange data via NFC with the attacker devices. However, the communication between the two attacker devices can be in an arbitrary way: They can use Bluetooth, a local WLAN connection or they can even be in two different countries and communicate via the internet [42].

A relay attack opens new ways of defrauding card holders: An attacker can now hold the counterfeit terminal near the victim's card (while being at e.g. a crowded public space) while another attacker pays with a connected counterfeit card at a remote shop. This works up to the limit where the card holder would be required to enter their PIN or another card holder verification has to be applied.

In theory the scenario of a crowded public space can also be exploited by a portable terminal that the attacker holds directly near the card of the victim without requiring the relay architecture. However, terminals have to be officially registered and the attacker could be easily detected when a victim issues a complaint (see subsection V-D).

The paper [42] describes a practical implementation of this relay attack: The authors wrote two mobile applications for the counterfeit card and terminal respectively. As hardware for those two devices, they used only commercially available mobile phones. They successfully tested the relay attack against both a test payment and e-passport system.

Another group executed a relay attack on payment terminals of public transport. They used a similar setup with two mobile phones working as relaying devices [43].

Paper [44] shows that a relay attack is also applicable over long distances by using two phones connected to a server via the mobile phone network. According to their time measurements, the attack was executed in less than 1 s in most cases. However, the group did not test the attack on wireless payment systems but with an e-passport. Relay attacks are therefore not a problem of the EMV protocol itself but they arise from the usage of the NFC technology.

There are solutions for the EMV protocol to prevent relay attacks: The forwarding of the messages needs some time that can be especially detected when the computation time is low. ToF? There have been countermeasures included in EMV against replay attacks but they differ between the payment providers. In general messages are sent by the terminal where the time for the response of the card is measured and if it is above a threshold, the transaction is aborted. However, as of 2022, there are still security problems with the implementation of relay protections or the allowed time frames are too high. This is the case as the companies use different techniques to prevent replay attacks and they are often proprietary. The Visa method could still be circumvented in 2022 by using rooted smartphones, as pointed out in [45].

C. Making EMV more secure

Is the EMV standard secure? This question from the beginning can not be answered in a single way. From the viewpoint of the previous magnetic stripe cards, the smartcards with chip are an improvement in terms of security. It is not possible to clone modern chip cards because some cryptographic material can not be read from outside the card.

However, the EMV standard is not a single smartcard but a protocol that runs on various smartcards, terminals and also NFC-enabled smartphones. As we have shown in this paper, the EMV standard is subject to current research and there are new vulnerabilities as well as flaws that have been found several years ago and that have already been fixed. The multipurpose and compatible design of EMV makes it likely that this trend will continue. By allowing multiple cardholder verification methods or different card authentication methods, as described in section VI there is automatically a higher risk that one of the methods can be used in an unintended way. Another problem of assessing the security of the EMV standard are the long and vague protocol descriptions. Concrete implementations can vary between different companies and therefore the security may also differ.

However, there are steps into the direction of proven security of the EMV protocol. The paper [3] takes an interesting approach in using the protocol verifier Tamarin to automatically find flaws. The Tamarin prover is a program which is able to evaluate all possible states of a protocol and has also been used in examining the mobile network standard LTE. The EMV standard can be studied this way by translating all human readable text of the specifications to Tamarin statements. The authors of [3] have found many known flaws with it and also discovered a new vulnerability: the MitM attack for the contactless Visa protocol (see subsection VII-A).

This can be a meaningful step into the direction of verifying correctness but it is not perfect so far. The authors admit that they only examined the implementations of the two largest companies MasterCard and Visa and other versions have not been verified.

Although there are several vulnerabilities in the EMV protocol discovered from time to time, the payments industry as a whole is arguably interested in the security. Therefore fixes are applied quickly after real-world exploits happen as could be seen in the case of the manufactured cards with an included Man in the Middle (see subsection VII-A). The EMV protocol includes no proof of absolute security. However, because all transaction data is stored in the banking networks, large scale attacks on EMV can arguably be noticed quickly.

References

[1] S. Schütz. "For many germans, cash is still king." (2019), [Online]. Available: https://www.npr.org/2019/06/09/728323278/for-many-germans-cash-is-still-king (visited on 02/07/2023).

[2] *Introduction to emvco*, 2022. [Online]. Available: https://www.emvco.com/wp-content/uploads/2022/08/EMV-2PP-Introduction_EURO_AUG22.pdf (visited on 12/01/2022).

[3] D. Basin, R. Sasse, and J. Toro-Pozo, "The emv standard: Break, fix, verify," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 1766–1781. DOI: 10.1109/SP40001.2021.00037.

[4] S. J. Murdoch, S. Drimer, R. Anderson, and M. Bond, "Chip and pin is broken," in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 433–446. DOI: 10.1109/SP.2010.33.

[5] "Capec-397: Cloning magnetic strip cards." (2021), [Online]. Available: https://capec.mitre.org/data/definitions/397.html (visited on 11/28/2022).

[6] J. Svigals, "The long life and imminent death of the mag-stripe card," *IEEE Spectrum*, vol. 49, no. 6, pp. 72–76, 2012. DOI: 10.1109/MSPEC.2012.6203975.

[7] D. Singh, P. Kushwaha, P. Choubey, A. Vaish, and U. Goel, "A proposed framework to prevent financial fraud through atm card cloning," in *Proceedings of the world congress on engineering*, vol. 1, 2011.

[8] D. Naccache and D. M'Raihi, "Cryptographic smart cards," *IEEE Micro*, vol. 16, no. 3, pp. 14–24, 1996. DOI: 10.1109/40.502402.

[9] G. Jain and S. Dahiya, "Nfc?: Advantages, limits and future scope," *vol*, vol. 4, pp. 1–12, 2015.

[10] N. E. Madhoun, E. Bertin, and G. Pujolle, "An overview of the emv protocol and its security vulnerabilities," in *2018 Fourth International Conference on Mobile and Secure Services (MobiSecServ)*, 2018, pp. 1–5. DOI: 10.1109/MOBISECSERV.2018.8311444.

[11] "Credit card." (2023), [Online]. Available: https://en.wikipedia.org/wiki/Credit_card (visited on 02/09/2023).

[12] *A guide to emv chip technology*, 2014. [Online]. Available: https://www.emvco.com/wp-content/uploads/2017/05/A_Guide_to_EMV_Chip_Technology_v2.0_20141120122132753.pdf (visited on 12/01/2022).

[13] N. Akinyokun and V. Teague, "Security and privacy implications of nfc-enabled contactless payment systems," in *Proceedings of the 12th international conference on availability, reliability and security*, 2017, pp. 1–10.

[14] EMVCo., Ed., *Book 1 Application Independent ICC to Terminal Interface Requirements*, EMV Integrated Circuit Card Specifications for Payment Systems, version 4.4, Oct. 2022.

[15] EMVCo., Ed., *Book 2 Security and Key Management*, EMV Integrated Circuit Card Specifications for Payment Systems, version 4.4, Oct. 2022.

[16] EMVCo., Ed., *Book 3 Application Specification*, EMV Integrated Circuit Card Specifications for Payment Systems, version 4.4, Oct. 2022.

[17] EMVCo., Ed., *Book 4 Cardholder, Attendant, and Acquirer Interface Requirements*, EMV Integrated Circuit Card Specifications for Payment Systems, version 4.4, Oct. 2022.

[18] EMVCo., Ed., *Book A Architecture and General Requirements*, EMV Contactless Specifications for Payment systems, version 2.10, Mar. 2021.

[19] EMVCo., Ed., *Book B Entry Point Specifications*, EMV Contactless Specifications for Payment systems, version 2.10, Mar. 2021.

[20] EMVCo., Ed., *Book D EMV Contactless Communication Protocol Specification*, EMV Contactless Specifications for Payment systems, version 2.6, Mar. 2016.

[21] EMVCo., Ed., *Book C-1 Kernel 1 Specification*, EMV Contactless Specifications for Payment systems, version 2.6, Feb. 2016.

[22] EMVCo., Ed., *Book C-2 Kernel 2 Specification*, EMV Contactless Specifications for Payment systems, version 2.10, Mar. 2021.

[23] EMVCo., Ed., *Book C-3 Kernel 3 Specification*, EMV Contactless Specifications for Payment systems, version 2.10, Mar. 2021.

[24] EMVCo., Ed., *Book C-4 Kernel 4 Specification*, EMV Contactless Specifications for Payment systems, version 2.10, Mar. 2021.

[25] EMVCo., Ed., *Book C-5 Kernel 5 Specification*, EMV Contactless Specifications for Payment systems, version 2.10, Mar. 2021.

[26] EMVCo., Ed., *Book C-6 Kernel 6 Specification*, EMV Contactless Specifications for Payment systems, version 2.10, Mar. 2021.

[27] EMVCo., Ed., *Book C-7 Kernel 7 Specification*, EMV Contactless Specifications for Payment systems, version 2.10, Mar. 2021.

[28] EMVCo., Ed., *Book C-8 Kernel 8 Specification*, EMV Contactless Specifications for Payment systems, version 1.0, Oct. 2022.

[29] J. van den Breekel, D. A. Ortiz-Yepes, E. Poll, and J. de Ruiter, "Emv in a nutshell," Tech. Rep., 2016.

[30] "Emvco members." (2021), [Online]. Available: https://www.emvco.com/about/emvco-members/ (visited on 11/28/2022).

[31] "Organization structure." (2021), [Online]. Available: https://www.emvco.com/about/organisation-structure/ (visited on 11/28/2022).

[32] "Emvco associates." (2021), [Online]. Available: https://www.emvco.com/about/overview/ (visited on 11/28/2022).

[33] "Pci security standards council about us." (2022), [Online]. Available: https://www.pcisecuritystandards.org/about_us/# (visited on 12/13/2022).

[34] "Emvco overview." (2021), [Online]. Available: https://www.emvco.com/about/overview/ (visited on 11/28/2022).

[35] "Was ist ein x.509-zertifikat?" (2019), [Online]. Available: https://www.ssl.com/de/faq/Was-ist-ein-x-509-Zertifikat%3F/ (visited on 02/07/2023).

[36] "Emv ca emv certificate authority." (2022), [Online]. Available: https://www.cryptomathic.com/products/emv/emv-ca (visited on 12/10/2022).

[37] "Emv key management explained, A white paper by cryptomathic," Tech. Rep., 2017. [Online]. Available: https://www.cryptomathic.com/whitepapers/emvkeymanagementexplained (visited on 12/10/2022).

[38] B. Latge. "What are emv® level 1 and level 2 testing?" (2021), [Online]. Available: https://www.emvco.com/knowledge-hub/what-are-emv-level-1-and-level-2-testing/ (visited on 02/07/2023).

[39] B. Latge. "What is level 3 terminal integration testing?" (2022), [Online]. Available: https://www.emvco.com/knowledge-hub/what-is-level-3-terminal-integration-testing/ (visited on 02/07/2023).

[40] ISO 7816-5:2004(E), "Identification cards — Integrated circuit cards — Registration of application providers," International Organization for Standardization, Geneva, CH, Standard, Apr. 2004.

[41] H. Ferradi, R. Géraud, D. Naccache, and A. Tria, "When organized crime applies academic results," in *IACR Cryptology ePrint Archive*, 2015, p. 20. [Online]. Available: https://eprint.iacr.org/2015/963.pdf.

[42] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, *Practical relay attack on contactless transactions by using nfc mobile phones*, Cryptology ePrint Archive, Paper 2011/618, https://eprint.iacr.org/2011/618, 2011. [Online]. Available: https://eprint.iacr.org/2011/618.

[43] T. Bocek, C. Killer, C. Tsiaras, and B. Stiller, "An nfc relay attack with off-the-shelf hardware and software," in *Management and Security in the Age of Hyperconnectivity*, R. Badonnel, R. Koch, A. Pras, M. Drašar, and B. Stiller, Eds., Cham: Springer International Publishing, 2016, pp. 71–83, ISBN: 978-3-319-39814-3.

[44] L. Sportiello and A. Ciardulli, "Long distance relay attack," in *Radio Frequency Identification*, M. Hutter and J.-M. Schmidt, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 69–85, ISBN: 978-3-642-41332-2.

[45] A.-I. Radu, T. Chothia, C. J. Newton, I. Boureanu, and L. Chen, "Practical emv relay protection," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 1737–1756. DOI: 10.1109/SP46214.2022.9833642.