dissertation final

by Mahamood Lidheem

Submission date: 14-Sep-2023 10:35AM (UTC+0100)
Submission ID: 212545909
File name: 32530_Mahamood_Lidheem_dissertation_final_511263_994060701.docx (14.3M)
Word count: 34829
Character count: 164542



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David
SWANSEA - ABERTAWE



School of
Applied Computing

INVESTIGATING EMV CHIP AND PIN CARD SECURITY

By

Mahamood Lidheem

Supervisor: Carlene Campbell

Project submitted as part of the requirements for
the award of MSc Computer Networks
and cyber-Security

14 September 2023

I. DECLARATION OF ORIGINALITY

I am Mahamood Lidheem, I declare that I am the sole author of this Project. that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed: MAHAMOOD LIDHEEM Date: 14/09/2023

II. FORM OF CONSENT

I am Mahamood Lidheem hereby consent that my Project, submitted in candidature for the M.Sc. Computer Networks and Cyber Security, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed: MAHAMOOD LIDHEEM Date: 14/09/2023

III. COPYRIGHT ACKNOWLEDGEMENT

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to University of Wales Trinity Saint David, Swansea.

IV. ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my project supervisor, Dr. Carlene Campbell, for her unwavering support and guidance throughout my entire master's degree course. Dr. Carlene has played a pivotal role in motivating me to strive for excellence and has provided invaluable assistance in achieving my goals. Her continuous support, including the provision of necessary equipment, has been instrumental in the successful completion of the proposed project.

Furthermore, I would like to express my gratitude to my classmates and university lecturers for their helpfulness and support over the past two years. Their collaboration and assistance have contributed significantly to my overall learning experience and personal growth.

Once again, I extend my heartfelt thanks to all those mentioned for their invaluable contributions and support throughout my university journey.

V. ABSTRACT

Following the COVID-19 outbreak in 2019, there was a global shift towards card payments over traditional cash transactions, preferring chip and pin cards. Within the context of this study, these are called EMV cards. In the financial industry, the EMV protocol is a global standard for credit and debit card transactions using chip enabled smart cards. EMV stands for Europay, MasterCard, and Visa, the three companies that initially developed the standard. As the adoption of these cards rose, so did the incidents of hacking and exploitation attempts. Motivated by these challenges, the author chose to investigate the security aspects of the EMV protocol.

Throughout the investigation, the author successfully executed a MITM attack on three bank cards, effectively intercepting the communications between the cards and the terminal. By carefully analyzing these intercepted packets, the author was able to extract crucial card details, including the card number, expiry date, card type, transaction amount, and currency. The author also managed to obtain the PIN code for two of the cards successfully

This report aims to achieve a deeper understanding of the EMV protocol's functioning and identify the specific protocol features employed in Chip-and-PIN cards. Additionally, the study seeks to identify any potential security vulnerabilities in the transaction process.

The findings and insights gained from this study are expected to contribute valuable knowledge to enhance the overall security of card payment systems in today's interconnected financial landscape.

Table of Contents

I. DECLARATION OF ORIGINALITY 2
II. FORM OF CONSENT 2
III. COPYRIGHT ACKNOWLEDGEMENT 2
IV. ACKNOWLEDGEMENTS 2
V. ABSTRACT 4
LIST OF FIGURES 7
LIST OF TABLES 8
TABLE OF ACRONYMS 8
1 INTRODUCTION 10
1.1 BACKGROUND 10
1.2 AIM 11
1.3 OBJECTIVES 11
1.4 STRUCTURE OF THE PROJECT 12
2 RESEARCH & REVIEW OF LITERATURE 14
2.1 INTRODUCTION 14
2.2 THE EMV COMPANY: INNOVATIONS AND SPECIFICATIONS 17
2.3 EMV CARD STRUCTURE AND CHARACTERISTICS 21
2.4 MAGNETIC STRIPE CARDS VS EMV CARDS 23
2.5 JAVA CARD 27
2.6 PAYMENT PROCESS (EMV PROTOCOL) 28
2.6.1 CARD INSERTION AND APPLICATION SELECTION 29
2.6.2 CARD VERIFICATION 31
2.6.3 EMV CHIP AUTHENTICATION 31
2.6.4 CARDHOLDER VERIFICATION 32
2.6.5 TRANSACTION AUTHORIZATION (REQUEST, RESPONSE) 34
2.6.6 TERMINAL RISK MANAGEMENT 40
2.6.7 GENERATE CRYPTOGRAM 41
2.7 TRANSACTION COMPLETION 42
2.8 ATTACKS ON EMV CHIP AND PIN CARDS 43
3 RESEARCH METHODOLOGY 46
3.1 INTRODUCTION 46
3.2 RESEARCH OVERVIEW 46
3.2.1 RESEARCH PHILOSOPHY 47
3.2.2 RESEARCH APPROACH 47
3.2.3 METHODOLOGICAL CHOICES 48
3.2.4 RESEARCH STRATEGIES 48

3.2.5 TIME HORIZONS 49
3.2.6 DATA COLLECTION METHODS 49
3.3 RESEARCH DELIVERABLES 51
3.4 CONCLUSION 51
4 DESIGN & IMPLEMENTATION 53
4.1 INTRODUCTION 53
4.2 PROBLEM FORMULATION 53
4.2.1 TOOLS AND PREREQUISITES 54
4.3 PROJECT DESIGN 57
4.3.1 PRE-EXPERIMENT DESIGN 57
4.3.2 EXPERIMENT DESIGN 58
4.4 IMPLEMENTATION 58
4.4.1 UBUNTU 22.04.2 LTS 58
4.4.2 OSMOCOM SIMTRACE 2 59
4.4.3 WIRESHARK PACKET SNIFFER 62
4.4.4 HARDWARE IMPLEMENTATION 65
4.4.5 SUMMARY 66
5 TESTING AND EVALUATION 67
5.1 INTRODUCTION 67
5.2 TESTING RESULTS DISCUSSION 67
5.2.1 TEST ONE, PRE-EXPERIMENT TEST 67
5.2.2 TEST TWO, SNIFFING BANK DEBIT CARDS DURING REALTIME TRANSACTION 69
5.2.3 SUMMARY 93
6 CONCLUSIONS AND RECOMMENDATIONS 95
6.1 SUMMARY 95
6.2 CONCLUSION 95
6.3 RECOMMENDATIONS FOR FUTURE RESEARCH 96
7 REFLECTIONS 97
7.1 INTRODUCTION 97
7.2 LESSON LEARNT 97
7.3 THINGS THAT COULD BE DONE DIFFERENTLY 97
7.4 BARRIERS EXPERIENCED 98
7.5 POSITIVES 98
7.6 GUIDANCE TO THE FUTURE STUDENTS 99
7.7 SUMMARY 99
8 REFERENCES 100

APPENDICES 103

LIST OF FIGURES

Figure 1: EMVCO owners 17
Figure 2: EMV standards 17
Figure 3: Front of EMV card [13] 21
Figure 4: Back of EMV card [13] 21
Figure 5: Chip electric contacts 23
Figure 6: EMV Card-Present Transaction Percentage [18] 26
Figure 7: Worldwide EMV Chip Card Deployment [18] 26
Figure 8: EMV Deployment Map [18] 27
Figure 9: Steps for EMV Contact Transaction [20] 29
Figure 10: Card insertion and Application selection 30
Figure 11: Command APDU Structure [22] 35
Figure 12: Response APDU Structure [23] 35
Figure 13: Most Significant Nibble of the Class Byte [23] 36
Figure 14: Coding of the Instruction Byte [23] 37
Figure 15:Structural Scheme of Status Bytes [23] 39
Figure 16: Coding of Status Bytes SW1 SW2 [23] 40
Figure 17: ATM skimmer in action 43
Figure 18: card Shimming 44
Figure 19: card reader Shimming 44
Figure 20: Research Onion [25] 46
Figure 21:SIMtrace2 Hardware Kit 54
Figure 22:SIMtrace microSIM (3FF) FPC Cable 55
Figure 23:SIMtrace Smart Card FPC 55
Figure 24:SIM card to FPC adapter 56
Figure 25: Card Machine 56
Figure 26: pretest design 57
Figure 27: Experiment Design 58
Figure 28: Ubuntu 22.04.2 LTS 59
Figure 29: Osmocom SIMtrace 2 60
Figure 30: Osmocom SIMtrace 2 60
Figure 31: Osmocom SIMtrace 2 61
Figure 32: Osmocom SIMtrace 2 61
Figure 33: Osmocom SIMtrace 2 62
Figure 34: Wireshark 63
Figure 35: Wireshark 64
Figure 36: Experiment pretest 65
Figure 37: Experiment test 65
Figure 38: wireshark simcard sniffing. 67
Figure 39:simtrace2 sniffing. 68
Figure 40: bank card sniffing 69
Figure 41: bank card sniffing Wireshark. 70
Figure 42: scapy python ..... **Error! Bookmark not defined.**
Figure 43:scapy results ..... **Error! Bookmark not defined.**
Figure 44: pyusb code ..... **Error! Bookmark not defined.**
Figure 45: Gantt chart original 1 117
Figure 46: Gantt chart original 2 117

Figure 47: Gantt chart actual 1 118
Figure 48: Gantt chart actual 2 118

LIST OF TABLES

Table 1: Data Collection Methods 50
Table 2: Intercepted Card Details from MITM Attack 93
Table 3: cardholder name 107

TABLE OF ACRONYMS

EMV	Europay, MasterCard and Visa
APDU	Application Protocol Data Unit
PIN	Personal Identification Number
MITM	Man-In-The-Middle
SDA	Static Data Authentication
DDA	Dynamic Data Authentication
CDA	Combined Data Authentication
ATM	Automated Teller Machine
CNP	Card-Not-Present
POS	Point-Of-Sale
ISO	International Organization for Standardization
SRC	Secure Remote Commerce
RF	Radio Frequency
QR	Quick Response
ICC	Integrated Circuit Card
PAN	Primary Account Number
CVV	Card Verification Value
CVC	Card Verification Code
DES	Data Encryption Standard
AES	Advanced Encryption Standard
RSA	Rivest-Shamir-Adleman
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
CVM	Cardholder Verification Method
ATR	Answer To Reset
RST	Reset
EPB	Encrypted PIN Block
C-APDU	Command Application Protocol Data Unit
R-APDU	Response Application Protocol Data Unit
CLA	Class
INS	Instruction
DOL	Data Object Lists
AVS	Address Verification Service
GAC	Generate AC
AAC	Application Authentication Cryptogram
TC	Transaction Certificate

ARQC	Authorization Request Cryptogram
PCB	Printed Circuit Board
FPC	Flexible Printed Circuit
LTS	Long-Term Support
PSE	Payment System Environment
FCI	File Control Information
SFI	Short File Identifier
AID	Application Identifier
RID	Registered Application Provider Identifier
PIX	Proprietary Identifier Extension
AVN	Application Version Number
GPO	Get Processing Options
CD	Cryptogram Data
AIP	Application Interchange Profile
AFL	Application File Locator
TLV	Tag, Length, Value
IAC	Issuer Action Code
CDOL	Card Risk Management Data Object List
CID	Cryptogram Information Data
ATC	Application Transaction Counter
IAD	Issuer Application Data

1 INTRODUCTION

1.1 BACKGROUND

EMV [1], as the world's leading standard for payments with smart cards, is originally named after its founders (Europay, MasterCard and Visa) in the 1990s. It aims to improve payment security and make it more difficult for fraudsters to successfully counterfeit cards. EMV card is also called Chip and PIN, which describes the process of credit and debit card transactions using EMV compliant microprocessor (chip) cards at a payment terminal and requiring the cardholder's PIN to be entered for verification. According to the statistics published on the EMVCo website [2], the number of EMV cards in use is over 10.816 billion by now, and worldwide almost all bank cards with a chip on them are EMV compliant. The EMV specification does not define a single protocol, but a highly configurable toolkit for payment protocols, which allow different combinations of choices among three card authentication methods (SDA, DDA, CDA), five cardholder authentication methods (none, signature, online PIN, offline plaintext PIN, and offline encrypted PIN), and two transaction types (online and offline) [3].

These cards are still susceptible to man-in-the-middle (MITM) attacks, which can compromise the security of the transaction. In this research the author tested the security of EMV chip and PIN cards against MITM attacks and evaluate the effectiveness of current security measures in preventing them.

One of the most common types of attacks on EMV chip and PIN cards is skimming. Skimming involves the use of a small device that is attached to a payment terminal or ATM to capture the data from the card's magnetic stripe or chip. Once the data is captured, it can be used to create a counterfeit card that can be used for fraudulent transactions. Skimming attacks can be difficult to detect, as the payment terminal or ATM may still appear to be functioning normally.

Another type of attack that is frequently used against EMV chip and PIN cards is cloning. Cloning involves the replication of the card data onto another card, which can then be used for fraudulent transactions. Cloning attacks are typically more difficult to carry out than skimming attacks, as they require more sophisticated equipment and skills. However, they can still be carried out successfully by determined attackers.

The vulnerabilities of EMV chip and PIN cards can be attributed to several factors. One of the main factors is the complexity of the EMV protocol itself. The EMV protocol is a complex

system that involves multiple layers of security, each of which can be targeted by attackers. Additionally, the EMV protocol is not a standardized system, and different implementations of the protocol can have varying levels of security.

Another factor that contributes to the vulnerability of EMV chip and PIN cards is the lack of awareness among cardholders and merchants about the risks associated with these cards. Many cardholders and merchants assume that EMV cards are completely secure and do not take adequate measures to protect themselves against potential attacks.

1.2 AIM

The main aim of this study is to Investigate the security mechanism employed by EMV chip and pin card. The findings will contribute to the ongoing efforts in enhancing the security of payment systems and may assist in the development of improved security measures for future iterations of EMV technology.

1.3 OBJECTIVES

This study has the following objectives:

1. Analysing the security mechanisms employed by the EMV protocol, including encryption, digital signatures, and authentication, to determine their effectiveness in protecting payment card data.
2. Identify and analysing common attack vectors used by attackers to exploit vulnerabilities in the EMV protocol, such as card skimming, man-in-the-middle attacks, and social engineering.
3. Evaluating the effectiveness of current countermeasures against attacks on the EMV protocol, including transaction limits, PIN entry timeouts, and online/offline PIN verification methods.
4. Developing additional countermeasures to improve the security of the EMV protocol, such as dynamic CVV codes, biometric authentication, and transaction monitoring systems.
5. Proposing recommendations to payment card issuers and terminal manufacturers to enhance the security of the EMV protocol, such as improving security standards and protocols, educating users about security risks, and adopting new security technologies.

6. Demonstrating the effectiveness of the SimTrace2 device in identifying potential vulnerabilities in the communication between the card and the terminal during a transaction, such as unauthorized data exchange and malicious code injection.

1.4 STRUCTURE OF THE PROJECT

The project can be divided into three main components. Firstly, there is the theoretical groundwork, which forms the foundation of the project. This involves reviewing existing literature and previous studies on EMV Chip and PIN card security. By expanding the knowledge base, this theoretical work is crucial to the overall outcome of the project.

The second part focuses on the design and implementation phase. Here, the author outlined the problem formulation and described the design that was used for conducting a MITM attack on an EMV card. This section also covered the required software and hardware necessary to successfully complete the project.

Finally, the third part of the project entailed presenting the findings. This involved examining the captured information during the MITM attack. The results obtained from this experiment helped determine the outcome. Additionally, this section included a summary of the previous chapters, a final conclusion, and suggestions for future work or development.

CHAPTER ONE: This is the background section of the report which provides a concise overview, highlighting the purpose and emphasizing the essential objectives required to achieve a successful outcome.

CHAPTER TWO: Involves the process of gathering information from previous studies and academic papers. This comprehensive review enables the author to successfully complete the proposed project by acquiring valuable insights and building upon existing knowledge in the field.

CHAPTER THREE: Is based on a methodology that provides a clear pathway for achieving targets and successfully completing the project. This structured approach ensures that the necessary steps and processes are followed to effectively accomplish the desired objectives.

CHAPTER FOUR: Focuses on the design and implementation aspects of the proposed project, offering relevant details and insights. This includes outlining the specific design considerations and strategies employed, as well as describing the practical steps taken to bring the project to fruition.

CHAPTER FIVE: Involves describing the specific procedures followed, the data collected, and the analysis methods employed to assess the outcomes of the experiments. The information offered aims to provide a comprehensive understanding of the testing and evaluation process and its significance in the context of the project.

CHAPTER SIX: In conclusion, the project findings are summarized, and recommendations for future improvements are provided. The conclusion highlights the key insights and outcomes derived from the project, while the recommendations focus on suggesting areas for further enhancement or development. These recommendations serve as valuable guidance for future iterations of the proposed project, allowing for continuous improvement and better outcomes.

CHAPTER SEVEN: This chapter provides reflections based on the research findings and achievements. It offers a thoughtful analysis and interpretation of the results obtained from the research, highlighting key insights and accomplishments. These reflections help to deepen the understanding of the research outcomes and their significance, providing a valuable perspective on the project's contributions to the field of study.

CHAPTER EIGHT: This chapter includes reference details for each paper used in the research, acknowledging the efforts of past researchers. By providing proper citations and acknowledging the contributions of previous studies, the report recognizes and respects the work done by other researchers in the field. These reference details serve to give credit to the authors and sources that have influenced and informed the current report.

2 RESEARCH & REVIEW OF LITERATURE

2.1 INTRODUCTION

A credit card is a payment card that allows users to purchase goods and services, both online and offline, by providing credit services to customers. The convenience and globalization of credit cards have resulted in an increase in credit card transactions worldwide. Unfortunately, this has also led to a rapid growth in fraudulent activities.

In recent years, there has been a significant increase in credit card fraud, with losses estimated to be in the billions of dollars annually.

According to the report published by UK Finance for the year of 2021 [4]:

- Losses due to remote purchase fraud also called as card-not-present (CNP) fraud total is £210.5 million in the first six months of 2021 [4].
- Losses from lost and stolen card fraud total is to £35.1 million in the first six months of 2021 [4].
- Losses from counterfeit card fraud total is £2.6 million in the first six months of 2021 [4].

To combat this problem, various measures have been implemented, such as the EMV chip and PIN protocol. The protocol is designed to provide enhanced security by using cryptographic algorithms to authenticate the card and the terminal during a transaction. The chip on the card generates a unique cryptogram for each transaction, which is verified by the terminal to ensure that the card is genuine, and that the transaction is authorized.

Card fraud can take place both online and offline. When fraudulent activity happens in a face-to-face transaction, it is considered offline card fraud. Examples of offline card fraud include skimming, where a fraudster copies card information from a physical card, or a dishonest merchant who charges for goods or services that were not delivered.

On the other hand, when fraudulent activity occurs over the internet or other digital channels, it is considered online card fraud. Phishing is one type of online card fraud where a fraudster sends an email or message that appears to be legitimate but aims to trick the recipient into disclosing their card information. Other forms of online card fraud include hacking, where the fraudster accesses a database of card information, and card-not-present fraud, where the fraudster uses stolen card details to make online purchases without having the physical card.

The use of EMV chip and PIN cards has become widespread in modern day banking due to their ability to provide secure transactions. However, the security of the EMV protocol has come under scrutiny in recent years due to several vulnerabilities that have been discovered. EMV chip and PIN cards are designed to prevent fraud by using a combination of cryptographic algorithms and secure elements to protect the cardholder's sensitive information. However, attackers have found ways to bypass these security measures, leading to fraudulent activities such as card cloning, skimming, and man-in-the-middle attacks.

To address these vulnerabilities, several studies have been conducted in recent years to reevaluate the security of the EMV protocol and develop countermeasures to prevent attacks. These studies have used various techniques, such as hardware security modules, distance bounding protocols, and secure elements, to enhance the security of the EMV protocol.

In 2020, researchers T. Van Goethem, F. Missurka, and T. Piessens conducted a comprehensive study focusing on the security of contactless payment systems, specifically the EMV contactless protocol used in Europe and other regions. The study, titled "First Contact: New vulnerabilities in Contactless Payments," explores various attack scenarios like the relay attack and the ghost and replay attack. It evaluates the effectiveness of existing countermeasures against these attacks and introduces novel countermeasures such as distance bounding protocols and dynamic transaction verification to enhance the security of contactless payment systems. The study emphasizes the necessity for improved security measures to prevent fraudulent activities in contactless payment systems and highlights the importance of educating consumers about the associated risks. By providing valuable insights into the vulnerabilities of contactless payment systems and proposing solutions, this study contributes to the development of more secure payment systems that safeguard consumer interests [5].

In the paper "EMV Electronic Payment System and its Attacks: A Review" by E. M. El-Hadedy, S. M. Abdalla, and A. F. El-Yamany published in 2020, the authors provide an overview of the EMV payment system and examine various attacks, including skimming, replay, relay, and man-in-the-middle attacks. The study evaluates the effectiveness of existing countermeasures and proposes enhancements such as dynamic transaction verification, distance bounding protocols, and biometric authentication. It highlights the importance of user awareness and recommends multi-layered security measures for the overall security improvement of the payment system [6].

Another study, "Attack Tree for Modelling Unauthorized EMV Card Transactions at POS Terminals" by J.S. Yeo and H.S. Kim published in 2018, focuses on the security of EMV card transactions at point-of-sale (POS) terminals. The study identifies attack scenarios, including card skimming, man-in-the-middle attacks, and transaction data modification. The authors propose an attack tree model to analyze the associated risks and develop effective countermeasures. The study underscores the importance of developing robust security measures to prevent unauthorized transactions and protect the interests of consumers [7].

"Formal Analysis of the EMV Protocol Suite" by Joeri de Ruiter and Erik Poll presents a formal analysis of the EMV protocol suite, widely used in chip-based payment systems. The study utilizes a model-based approach to identify potential security weaknesses. The authors demonstrate how vulnerabilities in the protocol could lead to fraudulent transactions or compromise the system's security. They propose improvements such as formal verification techniques and stronger cryptographic algorithms to enhance the protocol's security [8].

The study titled "Examining Users' Understanding of Security Failures in EMV Smart Card Payment Systems" by Akeem Olowolayemo, Nafisat Adewale, Akram M. Zeki, and Zubair Ahmad published in 2019 explores users' understanding of security failures in EMV smart card payment systems. The study reveals that users have limited awareness of security risks, particularly in contactless payments. It identifies factors such as age, educational level, and income level that influence users' understanding of these risks. The authors recommend educational campaigns to increase user awareness and understanding [9].

The merits of research papers in this category are that they provide valuable insights into the security vulnerabilities and attack scenarios associated with EMV payment systems. They propose new countermeasures to enhance the security of such systems and inform the development of more secure payment systems that protect the interests of consumers. By highlighting the limitations of existing countermeasures and identifying potential vulnerabilities, these papers can help improve the overall security of EMV payment systems.

However, one potential demerit of research papers in this category is that they may be highly technical and difficult for non-experts to understand. They may require a deep knowledge of computer science, cryptography, and related fields to fully grasp the concepts presented in the papers. Additionally, some papers may be focused on specific regions or systems, which may

limit their applicability in other contexts. Finally, research papers in this category may be quickly outdated due to the rapid pace of technological advancements in payment systems.

2.2 THE EMV COMPANY: INNOVATIONS AND SPECIFICATIONS

EmvCo is the International Authority for Specifications and Processes in Payment Systems EmvCo, a collaborative venture between American Express, Discover, JCB, MasterCard, UnionPay, and Visa, oversees and preserves global specifications and associated procedures. In a concerted effort, these six major payment networks collaborate to jointly develop, maintain, and safeguard the EMV chip specifications, thereby ensuring continuous interoperability and compatibility across the global payment industry. Figure 1 shows the logos of founder's companies.



Figure 1: EMVCO owners

EMV transaction specifications are based on a variety of standards, each serving a distinct purpose. Figure 2 gives a concise overview of these specifications:

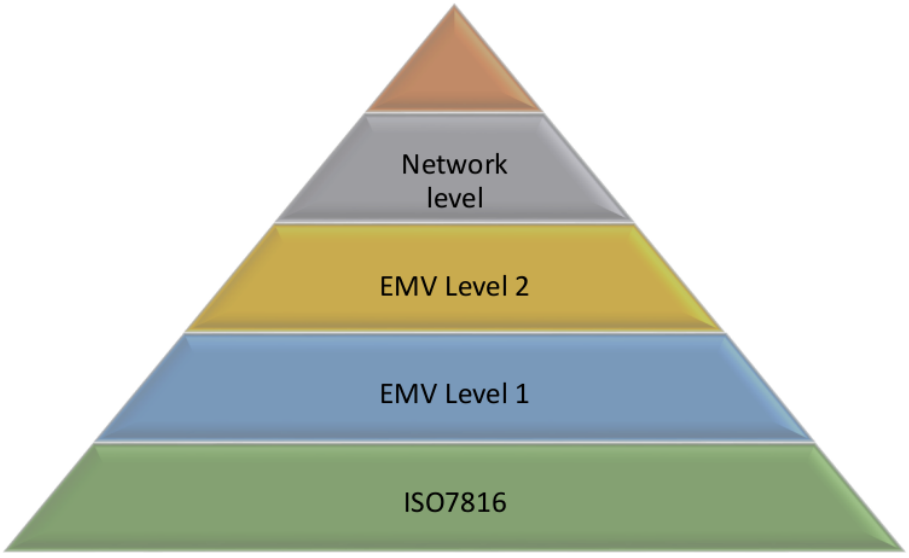


Figure 2: EMV standards

ISO 7816: Is a series of standards specifying integrated circuit cards and the use of such cards for interchange. These cards are identification cards intended for information exchange negotiated between the outside world and the integrated circuit in the card. As a result of an information exchange, the card delivers information (computation result, stored data) and modifies its content (data storage, event memorization) [10].

This standard operates at the lowest level, governing the fundamental workings of chip cards at the electronic level. It encompasses the card-terminal interface, command protocols, and common software functions shared by all chip cards.

These specifications are available for download from the ISO website, subject to a fee which is: CHF 145,00 equal to £129.48.

EMV Level One (Packet level): This specification focuses on the physical and electrical characteristics of payment cards. It covers card design, chip layout, contact arrangements, and the communication interface between the card and the terminal. It defines aspects such as voltage levels, timing requirements, and signal encoding for data transmission.

This standard is managed by EMVCO and are available for free of charge on the EMVCO website [11] .

EMV Level Two (Application Level): This specification ensures compatibility across all chip cards and facilitates interoperability among various payment networks. It establishes software requirements for payment applications on the cards, encompassing user interfaces, transaction processing, security features, and other relevant aspects. Its purpose is to enable any payment terminal worldwide to process transactions from any chip card, regardless of the payment network.

This standard is managed by EMVCO and are available for free of charge on the EMVCO website [11].

Network Level Specifications: These specifications pertain to payment networks and allow for differentiated applications and personalized features. They enable payment networks to manage risk according to their own strategies and define protocols for message formatting, data exchange, and authentication.

The management of Network Level Specifications varies depending on the specific payment networks. Each payment network typically has its own organization that oversees the development, maintenance, and governance of the network specifications. For example, Visa has Visa Inc (Visa Integrated Circuit Card Specifications)., MasterCard has Mastercard Incorporated (M/Chip 4 Card Application Specifications), and American Express has

American Express Company. These organizations work collaboratively with their respective network members and stakeholders to manage the Network Level Specifications and ensure their effective implementation within the payment network ecosystem.

In summary, the EMV specifications encompass ISO 7816 for low-level chip card operations, EMV level one for physical and electrical characteristics, EMV level two for application interoperability, and network-level specifications for network-specific functionalities and protocols.

After briefly explaining the EMV specification, the author discussed the current EMV chip contact specification V4.4, which was released in October 2022. This latest version incorporated numerous updates and enhancements compared to its predecessor. It included the following:

- **EMV chip contact specifications**: Used when the card is physically inserted into the terminal.
- **EMV chip contactless specifications**: Used when taping the card on the card machine.
- **EMV chip personalization specifications**: Used during the card manufacturing process.it defines how the card should be personalized.
- **EMV mobile specifications**: EMV mobile specifications refer to the guidelines and requirements set forth by EMVCo for enabling secure and standardized mobile payment transactions using mobile devices, such as smartphones or wearables. These specifications define the technical framework and protocols necessary to ensure interoperability and security in mobile payment solutions.

  The EMV mobile specifications cover various aspects, including tokenization, secure element management, transaction processing, authentication methods, and communication protocols. They aim to provide a consistent and secure experience for mobile payment users across different devices, operating systems, and payment networks.

  EMV standards and technology can used under three main categories:

  - **Face-to-face transactions (Card-present)**: Can be categorized into the following groups, each with their respective EMV specifications:

- **EMV Contact Specifications**: These specifications pertain to transactions where the chip card is physically inserted into a payment terminal, and the transaction is authorized through a contact interface.
- **EMV Contactless Specifications**: These specifications are relevant to transactions where the payment is made by tapping or waving a contactless-enabled card or device near a compatible payment terminal. The communication occurs via radio frequency (RF) technology.
- **EMV Mobile Specifications:** These specifications are specific to transactions conducted using mobile devices, such as smartphones or wearables, enabled with secure mobile payment technologies.
- **EMV QR Code Specifications**: These specifications govern transactions where a Quick Response (QR) code is scanned or displayed on a mobile device or payment terminal to initiate and authenticate the transaction.
- **EMV Wearable Device Specifications**: These specifications are tailored for transactions conducted using wearable devices, such as smartwatches or fitness trackers, which have embedded payment capabilities.

Each of these EMV specifications ensures secure, interoperable, and reliable payment transactions within their respective contexts, catering to diverse face-to-face transaction scenarios.

- **Remote transactions (e-commerce websites/Card-not-present):** EMV Secure Remote Commerce (SRC) specifications are designed for the secure and streamlined processing of e-commerce transactions. These specifications are utilized when conducting online purchases on e-commerce websites. They establish a standardized framework for secure payment acceptance, data protection, and authentication of the cardholder. In addition to EMV SRC, EMV 3-D Secure specifications are employed to authenticate the cardholder during online transactions. These specifications provide an extra layer of security by implementing a three-domain model (hence 3-D) for authentication, involving the payment issuer, the acquirer, and the cardholder.

2.3 EMV CARD STRUCTURE AND CHARACTERISTICS

The current credit card design is based on ISO 7810 [12] which provides specifications for different card sizes, including ID-1, which is the most common size for payment cards, measuring 85.60 mm × 53.98 mm (3.370 in × 2.125 in). This standard also covers card thickness variations, such as ID-1 CR80 cards, which are typically 0.76 mm (0.030 in) thick. Let's consider Figures (3,4) which shows the credit card elements.



Figure 3: Front of EMV card *[13]*

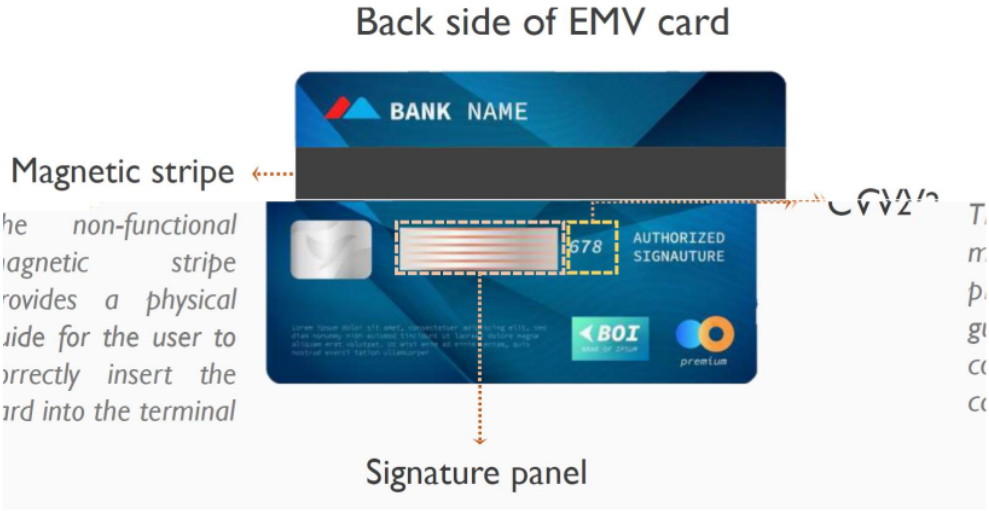


Figure 4: Back of EMV card *[13]*

1. **Card brand logo**: It's also called Card scheme branding.
2. **Card number**: It's also called PAN number and it's defined in ISO/IEC 7811 [14] international standard, the Primary Account Number (PAN), which is the unique identifier of the cardholder, is encoded on both the magnetic stripe and the chip, if the card has one. The magnetic stripe contains the PAN in a specific format that can be read by magnetic stripe card readers. Similarly, if the card is equipped with a chip, the PAN and other relevant card data are securely stored within the integrated circuit.
3. **Expiration date:** Typically, the format for representing the expiration date on payment cards consists of two digits indicating the month followed by two digits indicating the year. This format, representing the validity period of the card, is commonly used. Additionally, the expiration date is encoded on both the magnetic stripe and the chip.
4. **Cardholder name:** The name on a payment card is typically limited to a maximum of 26 characters. This ensures that the cardholder's name can be displayed in a human-readable format on the card itself. Additionally, the card contains the cardholder's name in a machine-readable format.
5. **EMV™chip:** Also called integrated circuit card (ICC) and must comply with the physical characteristics specified by ISO 7816-1 [10]. These characteristics encompass the size, shape, location, and number of contacts on the card. Different types of chips exist, such as memory chips and microcontroller chips. Let's explore the difference between them. Starting with the memory chip, it provides a secure storage solution for sensitive data and ensures the integrity of the stored information. It is primarily used for storing static data, such as cardholder details, application data, and issuer-specific data. However, a memory chip alone lacks the necessary processing capabilities to perform complex operations or cryptographic functions.

   On the other hand, the microcontroller chip offers intelligence and computational power. It enables the execution of complex operations, cryptographic functions, and transaction processing. By combining processing capabilities with memory, the microcontroller chip allows for the execution of application code and effective management of data stored in the memory chip. Additionally, there are different variants of microcontroller chips, some with a coprocessor and others without. The key distinction is that microcontroller chips with a coprocessor are capable of performing offline transactions. Figure 5 gives a concise overview of contact plate of the chip



Figure 5: Chip electric contacts

- **C1 - VCC (Power Supply)**: Provides power to the chip. The voltage levels can vary, but common levels are 1.8V, 3V, or 5V.
- **C2 - RST (Reset)**: Used to reset the chip's internal state to its starting conditions.
- **C3 - CLK (Clock)**: Provides the chip with a clock signal to synchronize the operations. The reader typically provides this clock signal, and the chip performs its operations based on this clock.
- **C4 - RFU (Reserved for Future Use)**: This contact is reserved for potential future uses and is typically not connected to any circuit in the current specifications.

- **C5 - GND (Ground)**: Electrical ground reference point.
- **C6 - VPP (Programming Voltage)**: Historically used for programming the chip, but in modern EMV cards, this is often not connected since it's not needed for standard operation. It's a relic from older memory card standards.
- **C7 - I/O (Input/Output)**: Data line for bidirectional communication between the chip and the card reader.
- **C8 - RFU**: Like C4, this contact is also reserved for potential future use.

2.4 MAGNETIC STRIPE CARDS VS EMV CARDS

The magnetic stripe (often called magstripe) is a band of magnetic material containing iron-based particles capable of storing data. The data is read when the card is swiped, when the magnetic stripe is read by a magnetic head. Physical-level details of the data-storing method are described in the ISO/IEC 7811 standard [15].

The original magnetic stripe card relied on a method invented by Forrest Parry, an IBM engineer back in 1960s, who attached the magnetic stripe to plastic using a heating process [15].

The magnetic stripe on a payment card, such as a credit or debit card, typically consists of three tracks: Track 1, Track 2, and Track 3. The formats for Track 1 and Track 2 are defined in the ISO/IEC 7813 standard [16], while Track 3 is described in the ISO/IEC 4909 standard [17].

Track 1 and Track 2 are primarily used for reading data from the magnetic stripe and are generally read-only. They contain essential information such as the Primary Account Number (PAN), the cardholder's name, the card's expiration date, and discretionary data. Discretionary data is specific to each card issuer and may include additional information encoded at various offsets.

Alphanumeric data on the magnetic stripe, such as the cardholder's name, is encoded using the ANSI/ISO ALPHA Data Format. Numeric data, such as the PAN and expiration date, is encoded using the ANSI/ISO BCD Data Format. These formats define how the data is structured and stored on the magnetic stripe.

Track 3, on the other hand, has the ability to be updated by a terminal. However, this track is rarely used by issuers, especially with the introduction of integrated chip (EMV) technology and related standards. The capabilities of Track 3 are not widely utilized in modern payment card systems.

Regarding the discretionary data on Tracks 1 and 2, it can include various values that differ between issuers. This data may contain a cryptographic signature known as the Card Verification Value (iCVV or CVV1) or Card Verification Code (CVC). The CVV1 serves as a means of validating the card's authenticity and is usually generated using an algorithm and a secret key known only to the issuer. When a transaction occurs, the CVV1 is often sent to the payment network and validated by the issuer. CVV1 is different from CVV2 which is printed on the back of the card.

Magnetic stripe cards and EMV cards are two different types of payment cards with varying levels of security and functionality. Here's a comparison between the two:

1. **Magnetic Stripe Cards:**
    - Magnetic stripe cards have a black magnetic stripe on the back containing the cardholder's information.
    - They store data magnetically and require the card to be swiped through a card reader.
    - The data on the magnetic stripe is static and can be easily copied or cloned, making these cards susceptible to fraud and counterfeit activities.
    - Magnetic stripe cards are widely used in many countries, but their usage is gradually declining due to security concerns.
2. **EMV Cards:**
    - EMV cards, also known as chip cards or smart cards, have an embedded microchip on the front that stores and processes data.
    - Instead of swiping, these cards are inserted into a card reader or used for contactless payments.
    - The microchip generates a unique transaction code for each transaction, making it extremely difficult to clone or counterfeit.
    - EMV cards provide enhanced security compared to magnetic stripe cards, significantly reducing the risk of fraud, particularly in face-to-face transactions.
    - EMV technology has become the global standard for payment cards, and many countries have transitioned to EMV-based systems.

Key differences:

- **Security:** EMV cards offer superior security compared to magnetic stripe cards due to their dynamic transaction codes and chip-based authentication. Magnetic stripe cards are more vulnerable to cloning and counterfeiting.
- **Acceptance:** While EMV cards are widely accepted globally, magnetic stripe cards may face limitations, especially in regions that have migrated to EMV-based systems.
- **Liability:** In some regions, the liability for fraudulent transactions may shift to the party with the less secure technology. This often means that if a magnetic stripe card is used at a chip-enabled terminal, the issuer may bear the liability for any fraud that occurs.
- **Functionality:** EMV cards can support additional features like contactless payments, stronger authentication methods such as PIN, and integration with mobile payment systems.

Following the explanation of EMV cards and magnetic stripe cards, Figures (6,7,8) illustrates the global adoption and prevalence of EMV cards.

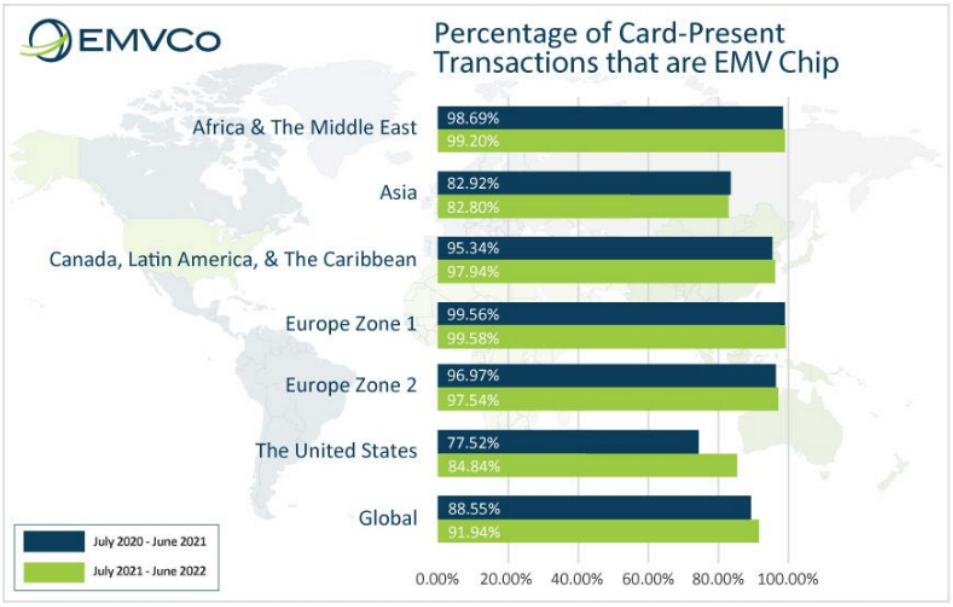


Figure 6: EMV Card-Present Transaction Percentage *[2]*

Worldwide EMV Chip Card Deployment and Adoption*

	2019		2020		2021	
Region	EMV Cards	Adoption Rate	EMV Cards	Adoption Rate	EMV Cards	Adoption Rate
Africa & the Middle East	312M	89.4%	339M	90.4%	375M	91.69%
Asia Pacific	6,226M	58.1%	6,885M	60.9%	7,528M	62.64%
Canada, Latin America, and the Carribbean	923M	86.7%	1,023M	90.7%	1,222M	90.56%
Europe Zone 1	1,040M	85.9%	1,073M	86.5%	1,192M	90.46%
Europe Zone 2	318M	80.7%	335M	84.1%	379M	86.29%
United States	1,074M	60.9%	1,161M	63.0%	1,282M	62.86%
Global	9,893M	63.8%	10,816M	66.4%	11,981M	68.16%

*The statistics show worldwide EMV Chip Card deployment figures as of Q4 2019, 2020, and 2021. The figures represent the latest statistics from American Express, Discover, JCB, Mastercard, UnionPay, and Visa, as reported by their member financial institutions globally.

Figure 7: Worldwide EMV Chip Card Deployment *[2]*

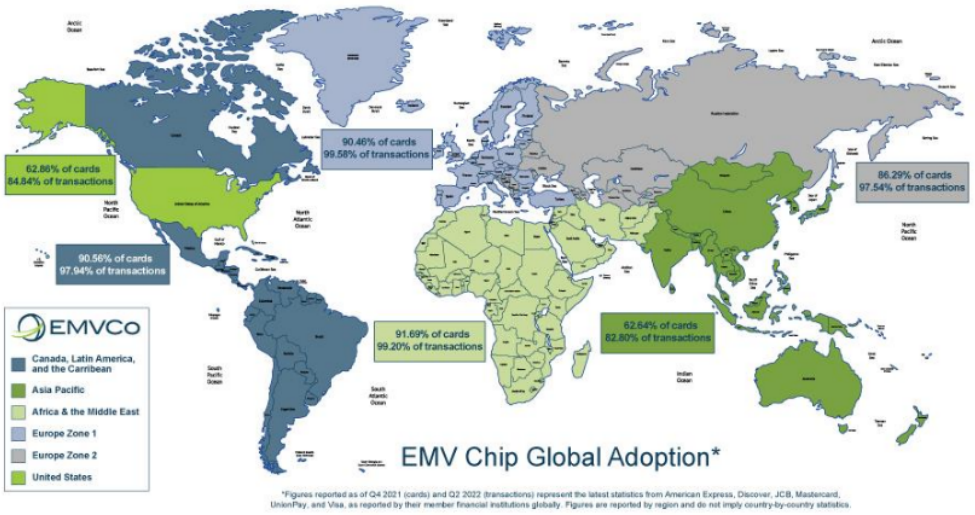


Figure 8: EMV Deployment Map *[2]*

2.5 JAVA CARD

Java Card is a technology that allows Java Card applets (small Java applications) to be run securely on smart cards. Billions of Java Cards are used as SIM cards, payment cards, ID cards, e-Passports, and more. Java Card applets are small programs, that can communicate with the terminal and with each other. The applet is a state machine which reads incoming commands, processes them, and responds back by sending data and status. Java card technology enables multiple applications to be deployed securely on a single smart card. Applets can be either pre-loaded on the card, or loaded into a smart card once it has been issued [18]. Java Card technology is often used in EMV cards to implement the necessary payment applications and security features.

The Java Card run-time environment conducts bytecode verification during the applet loading process. Before being loaded onto the card, the applet undergoes rigorous verification and execution. Once verified, the applet is signed and securely installed on the card. It's important to note that Java Card performs lightweight bytecode verification directly on the card. Once the applet is verified and signed, it becomes immutable, and any modifications would necessitate a fresh verification.

The primary goal of Java Card technology is to ensure the secure storage of sensitive information on smart cards. Java cards are equipped with hardware acceleration capabilities for various cryptographic operations, including symmetric cryptography (DES, Triple DES, AES), asymmetric cryptography (RSA, DSA, ECC), and other essential functions. Java Card's security foundation lies in the inherent robustness of the Java programming language, which provides a secure execution environment [18].

Sensitive data is stored securely within the application, and Java Card applications are executed within the isolated environment of the Java Card virtual machine, which operates separately from the underlying operating system and hardware.

In summary, the Java Card run-time environment ensures bytecode verification during applet loading, signing and installation of verified applets on the card, and the use of hardware acceleration for cryptographic functions to secure sensitive information. The isolated Java Card virtual machine and the applet firewall further safeguard data and applications from unauthorized access and modifications.

2.6 PAYMENT PROCESS (EMV PROTOCOL)

During the early days of bank card transactions, completing a transaction involved several steps. First, the customer would purchase goods from the merchant, and the merchant would call the bank for verification. If the card was not issued by the bank, the bank would then call the credit card company and look up the customer's name and credit balance. The process would then be reversed until the merchant received confirmation about the transaction.

However, with the introduction of magnetic stripe cards, the transaction process was greatly simplified. These cards store information such as an encrypted PIN, country code, currency units, and authorized amount using tiny iron-based magnetic particles in a plastic-like film. Despite their convenience, magnetic stripe cards are not very secure and have security issues. For example, merchants still need to verify the card, and in cases where transactions occur away from the terminal, such as in a restaurant, dishonest employees could take the card away and use a cheap machine to record the information on the card stripe. Additionally, even at the terminal, a thief could bend down in front of the customer and use a hidden reader to swipe the card, making illegal cloning of cards relatively easy and more common than before.

Due to the high cost of calling from Europe to the USA for transaction verification, an alternative verification method was needed. This led to the establishment of EMV by Europay, MasterCard, and Visa in 1993, which was later joined by Discover, JCB, UnionPay, and American Express to form EMVCo. The first EMV system was released in Europe in 1994.

The EMV card is classified as a smart card as it comprises of an integrated circuit that is embedded in accordance with ISO/IEC 7810 and ISO/IEC 7816 series of standards. The integrated circuit includes both memory and a general-purpose microcontroller. Java Card is often used as platform payment applications are then in the form of an applet.

The EMV protocol uses a chip and PIN system, where a small microprocessor chip is embedded in the card. When the card is used, the chip generates a unique code for each transaction. Figure 9 shows the steps of EMV Contact Transaction.

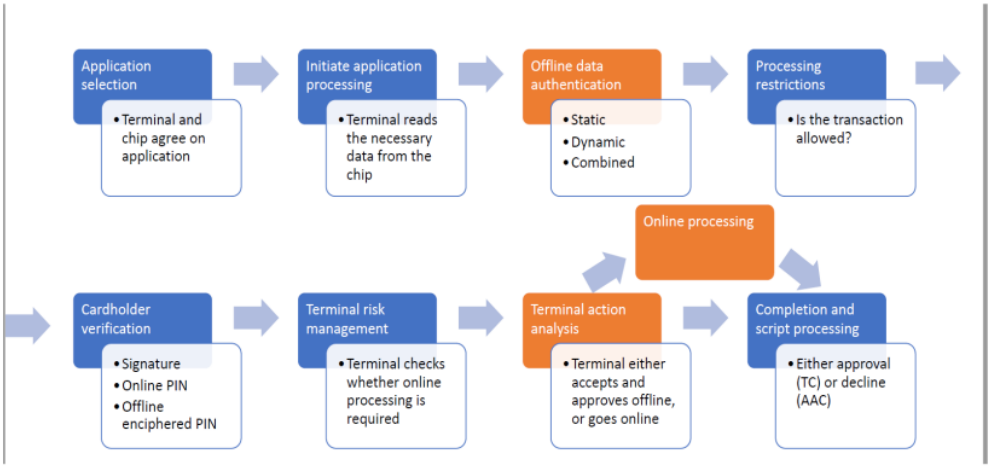


Figure 9: Steps for EMV Contact Transaction *[19]*.

2.6.1 CARD INSERTION AND APPLICATION SELECTION

The customer inserts the card into the POS terminal, and the chip on the card communicates with the terminal. The process known as technology selection involves the collaboration between the terminal, merchant, and cardholder to determine the method of data entry used during a transaction. Each EMV transaction involves the interaction between two applications: one hosted in the terminal (the terminal application) and the other hosted in the card (the card application). Cards do not have a built-in battery and rely entirely on the terminal for power.

Therefore, any communication, whether through a chip or contactless interaction, between the terminal and the card begins with the terminal supplying power to the card. In a contact-based environment, as illustrated in Figure 10, the terminal initiates communication by sending a signal on the card's RST line. The card then responds with a brief ATR (answer-to-reset) communication. From that moment until the transaction is completed, interrupted, or canceled, the terminal maintains control over the communication between itself and the card.

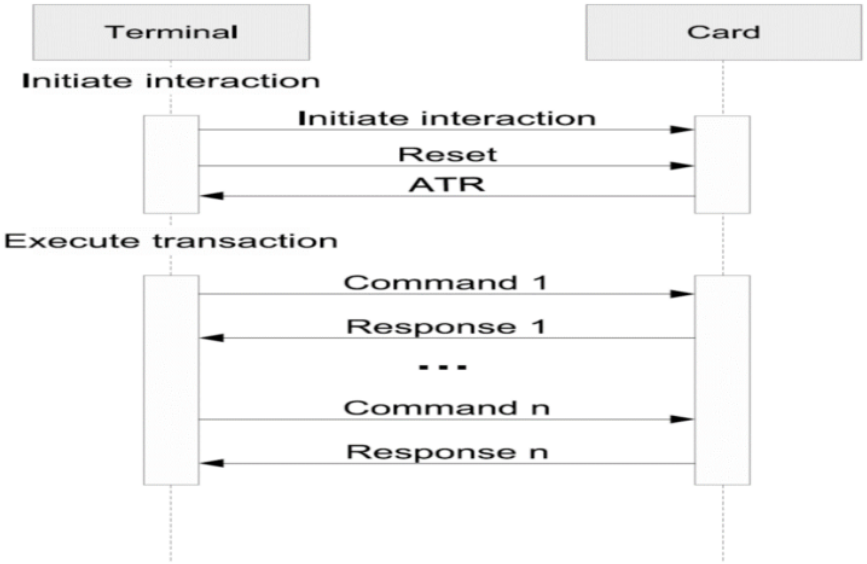


Figure 10: Card insertion and Application selection

The terminal application communicates with the ICC via physical contacts on the card. In a contactless scenario the communication and the power are provided wirelessly.

The ICC hosts a file system containing dedicated files and elementary files. Dedicated files can be referenced to via an application ID. Elementary file can be accessed by terminal application directly to retrieve necessary data from the ICC application. The terminal application sends commands (instructions) to the card application to which the latter responds with status codes and relevant data.

EMV specification defines a subset of ICC commands used for the interaction with an EMV application.

Furthermore, the EMV specification defines a tag-level-value format for its data, defines all supported tags and tag combinations (EMV templates) as well as adds the notion of DOLs (data

object lists) a way for terminal-card interaction, whereas the card informs the terminal of expected tags and their allocated lengths, and the terminal responds with the unformatted packed data [15].

2.6.2 CARD VERIFICATION

Card verification is the process of authenticating the validity and legitimacy of a payment card during a transaction. It involves various methods to ensure that the card being used is genuine and belongs to the authorized cardholder. Card verification is crucial in preventing fraudulent activities and unauthorized card usage.

There are several methods of card verification, including:

1. Card Verification Value (CVV/CVC): This method involves checking the three- or four-digit code printed on the back of the card. It adds an extra layer of security by verifying that the person making the transaction has physical possession of the card.
2. Address Verification Service (AVS): AVS compares the billing address provided during the transaction with the address on file with the card issuer. If there is a mismatch, it may indicate a suspicious transaction.
3. 3D Secure: This is an additional layer of security for online transactions. It typically involves the cardholder being redirected to a secure authentication page where they must enter a password, or a one-time verification code sent to their mobile device.
4. EMV Chip Authentication: EMV chips on cards use cryptographic algorithms to authenticate the card's validity. When inserted into a chip-enabled terminal, a unique transaction-specific code is generated, making it difficult for counterfeit cards to be used.

2.6.3 EMV CHIP AUTHENTICATION

Card Authentication Mechanisms (CAMs) The EMV standard defines three card authentication mechanisms: SDA, DDA, and CDA [8]:

- In SDA (Static Data Authentication) the card provides some digitally signed data (including e.g., the card number and expiry date) to the terminal to authenticate itself. This allows the terminal to check the authenticity of this data, since it knows the issuer's public key, but does not rule out cloning.

- DDA (Dynamic Data Authentication) cards can do asymmetric crypto and have a public/private key pair. DDA then involves a challenge-response mechanism, where the card proves its authenticity by signing a challenge chosen by the terminal using a private asymmetric key. Unlike SDA, this does rule out cloning. After the authentication of the card, DDA does not tie the subsequent transaction to that card. In other words, the terminal cannot verify that the transaction was actually carried out by the card.
- CDA (Combined Data Authentication) repairs this deficiency of DDA. With CDA the card digitally signs all important transaction data, not only authenticating the card, but also authenticating the transaction. The data that is authenticated with SDA, referred to in the standard as Static Data to be Authenticated, is also authenticated with DDA or CDA. For these authentication methods a hash over the data is included in the certificate containing the public key of the card. In this sense DDA and CDA subsume SDA [9].

2.6.4 CARDHOLDER VERIFICATION

Cardholder verification is the process of confirming the identity of the person using a payment card during a transaction. It is an essential step in ensuring that the individual attempting to make a purchase or perform a transaction is the authorized cardholder.

There are various methods of cardholder verification that are commonly used:

- **Failed CVM**: This method is used as a fail-safe or a fallback mechanism in exceptional situations. By designating failed CVM as the last rule, the card issuer can ensure that even if other verification methods or rules cannot be executed or are not applicable, the transaction will not proceed without successful verification. The purpose of this approach is to prioritize security and prevent unauthorized or fraudulent transactions from occurring in cases where the usual verification methods cannot be applied or have failed for some reason. It’s important to note that this specific implementation may vary depending on the card issuer, payment network, and the specific card's configuration. The inclusion of failed CVM as a deliberate rule is primarily for ensuring security and compliance with the card issuer's policies.

- **No CVM**: It means No Cardholder Verification Method. It refers to a specific scenario in card transactions where no cardholder verification is required or performed. In certain situations, particularly for low-value transactions or transactions that meet specific criteria set by the card issuer, no cardholder verification is needed. This means that the card can be used for payment without the need for the cardholder to enter a PIN, provide a signature, or undergo any other verification method. The No CVM method is typically employed to streamline and expedite transactions that are considered low risk or have a low transaction value. By skipping the cardholder verification step, the payment process can be faster and more convenient for both the cardholder and the merchant.

  It's important to note that the availability and application of No CVM may vary depending on the card issuer's policies, the transaction value, the merchant's capabilities, and the specific payment network involved. The goal is to balance convenience and efficiency while maintaining an acceptable level of security for eligible transactions.

- **Signature:** It is an older approach where a sales receipt is printed by the terminal. The customer is then requested to sign the receipt, and ideally, the employee should compare the signature on the receipt with the signature stripe on the card, as required by regulations. However, this method has significant limitations. Not only is it relatively easy to copy the signature already present on the card, but shop employee often ignores the signature comparison step. It's important to note that this verification method is not dependent on EMV technology and can be used with both magnetic stripe cards and EMV cards.
- **Offline PIN:** Can be further categorized into two types: plaintext offline PIN and enciphered offline PIN. In both cases, the integrated chip circuit embedded in the card is responsible for validating the PIN and verifying the identity of the cardholder. In the case of plaintext offline PIN, the entered PIN value is transmitted to the card without encryption. On the other hand, with enciphered offline PIN, as the name suggests, the PIN is transmitted to the card in an encrypted format. Terminals equipped with a PIN pad, which are certified and compliant with EMV standards, typically support both methods. The choice between plaintext and enciphered offline PIN validation is determined by the issuer and constrained by the computational capacity of the

integrated chip as enciphered PIN validation requires more computational resources. The encoding of PIN data exchanged between the integrated chip and the PIN pad is governed by the ISO/IEC 9564 standard [20]. This standard outlines the protocols and procedures for securely handling PIN data during the verification process. It's worth noting that the offline PIN verification method adds an additional layer of security to EMV transactions, ensuring that the cardholder's identity is validated using a personal identification number (PIN) entered on a secure PIN pad.

- **Online PIN:** The online PIN verification method involves encrypting the PIN as an encrypted PIN block (EPB) and transmitting it to the card issuer for validation, along with other transaction details. The EPB can have different formats as specified by the ISO/IEC 9564 standard [20]. The terminal encrypts the PIN block using the Triple Data Encryption Standard (3-DES). Once encrypted, the EPB undergoes multiple PIN translations as it passes through a terminal management system, including the host and payment networks. During these translations, the EPB is decrypted and re-encrypted using a symmetric secret key shared among the subsequent entities in the connection chain. This verification method, known as online PIN, is not exclusive to EMV technology and can also be used with magnetic stripe cards. It provides an added layer of security by ensuring that the PIN is securely packaged and transmitted to the issuer for validation, alongside other transaction details.

2.6.5 TRANSACTION AUTHORIZATION (REQUEST, RESPONSE)

During a transaction, the card and the terminal engage in a series of activities. First, the card generates a cryptographic signature using various transaction details, such as the amount, currency, date/time, and other relevant fields. This signature ensures the integrity and authenticity of the transaction data.

Simultaneously, the terminal plays its role by creating a payment message that encompasses additional business environment information, including the terminal ID, date, and time. The transaction-specific details, such as the amount and currency, are also included in this message. Additionally, the terminal encrypts the PIN and incorporates the chip data for security purposes.

Once the payment message is complete, it is forwarded to the payment network. The payment network acts as the intermediary, facilitating the communication between the terminal and the respective payment processing entities. This allows for further processing and authorization of the transaction.

Any application protocol is designed as a sequence of commands and corresponding command responses. Each of these messages is referred to as an Application Protocol Data Unit (APDU). When a command is initiated from the terminal application to the card application, it is called a Command APDU (C-APDU). On the other hand, the card application responds to the command with a Response APDU (R-APDU).

Card commands are categorized into different instruction classes. To specify a particular command, it is essential to transmit its command class, known as CLA, and the specific instruction code within that class, denoted as INS. Additionally, the protocol enables the inclusion of two single-byte parameters (P1 and P2) along with each command, and it allows for an additional variable-length field for supplementary parameters.

For each command, the Response APDU (R-APDU) comprises an optional variable-length data section, followed by a mandatory trailer. The trailer consists of two status bytes, SW1 and SW2, which indicate the outcome of the command, Figure 11 shows the structure of the command APDU, Figure 12 shows the structure of the response APDU as well.

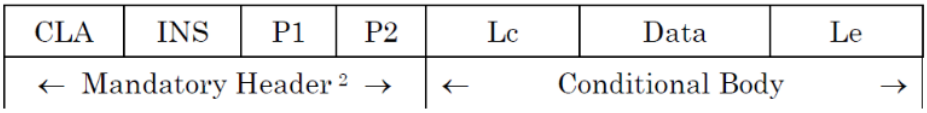


Figure 11: Command APDU Structure *[21]*

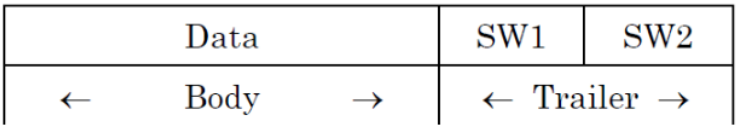


Figure 12: Response APDU Structure *[22]*

CLA: It is a mandatory 1-byte field in the Command APDU (C-APDU) that specifies the command class or logical channel of the command being sent to the card application.

The CLA byte is used to categorize commands into different instruction classes. Each instruction class represents a specific set of commands or functionality within the card application. By including the CLA byte, the sender can indicate which instruction class the command belongs to, Figure 13 shows the most significant nibble of the class byte.

The value of the CLA byte can vary depending on the card application and the specific protocol being used. Typically, the most significant 3 bits (bits 7-5) of the CLA byte are reserved for indicating the command structure and security-related information. The remaining 5 bits (bits 4-0) are available for application-specific use, allowing for up to 32 instruction classes within the protocol.

Hex	Meaning
'0'	Inter-industry command
'8'	Proprietary to this specification
Any other value	Outside the scope of this specification

Figure 13: Most Significant Nibble of the Class Byte *[22]*

INS: It is a mandatory 1-byte field in the Command APDU (C-APDU) that specifies the specific instruction or command within the command class.

The INS byte allows differentiation between different commands within the same instruction class. Each command represents a particular operation or action that the card application should perform in response to the received command.

The value of the INS byte is specific to the card application and the underlying protocol being used. Different instructions may have different meanings and effects on the card application, Figure 14 shows the meaning of the different instruction byte.

CLA	INS	Meaning
'8x'	'1E'	APPLICATION BLOCK
'8x'	'18'	APPLICATION UNBLOCK
'8x'	'16'	CARD BLOCK
'0x'	'82'	EXTERNAL AUTHENTICATE
'8x'	'AE'	GENERATE APPLICATION CRYPTOGRAM
'0x'	'84'	GET CHALLENGE
'8x'	'CA'	GET DATA
'8x'	'A8'	GET PROCESSING OPTIONS
'0x'	'88'	INTERNAL AUTHENTICATE
'8x'	'24'	PERSONAL IDENTIFICATION NUMBER (PIN) CHANGE/UNBLOCK
'0x'	'B2'	READ RECORD
'0x'	'A4'	SELECT
'0x'	'20'	VERIFY
'8x'	'Dx'	RFU for the payment systems
'8x'	'Ex'	RFU for the payment systems
'9x'	'xx'	RFU for manufacturers for proprietary INS coding
'Ex'	'xx'	RFU for issuers for proprietary INS coding

Figure 14: Coding of the Instruction Byte *[22]*.

P1 P2: The parameter bytes P1 P2 may have any value. If not used, a parameter byte has the value '00' [22].

Lc: Is an optional length field within the Command APDU (C-APDU) that indicates the number of bytes present in the subsequent data field of the command. Its purpose is to specify the length of the data being transmitted.

As per the ISO/IEC 7816-4 standard, the Lc field can be either 1 byte or 3 bytes long, providing flexibility in specifying a wide range of data lengths. However, in EMV applications, only 1-

byte length values are used. Therefore, in EMV applications, the Lc field will always consist of a single byte.

If the Lc field is included in the C-APDU, it must be immediately followed by the corresponding data field, which contains the actual command data. The length indicated in the Lc field should match the actual length of the data field. The card application interprets the Lc field to determine the size of the incoming data and processes it accordingly.

Data: The Data Field refers to a variable-length field within the Command APDU (C-APDU) or Response APDU (R-APDU) that contains the actual data being transmitted or received.

In the C-APDU, the Data Field follows the Command Class (CLA), Instruction Code (INS), and optional parameters (P1, P2) if applicable. It carries the specific data associated with the command being sent from the terminal application to the card application.

In the R-APDU, the Data Field contains the response data provided by the card application in response to the received command. The length and content of the Data Field depend on the specific command and the application protocol being used.

The Data Field is variable-length, meaning it can vary in size depending on the amount of data being transmitted or received. The length of the Data Field is often indicated by the Length (Lc) field in the C-APDU or obtained from the structure of the response in the R-APDU.

Le: Stands for "Expected Length" and is an optional field within the Command APDU (C-APDU) that specifies the expected maximum length of the Response APDU (R-APDU) data field.

In other words, optional length field of up to 3 bytes specifying the maximum length of the response data field in bytes expected for the command. The EMV standard specifies 1 byte as the field maximum length. If the field is present and equal to zero, then the terminal is ready to receive the maximum (256) bytes of data in the response APDU. If a C-APDU contains a single byte in the optional body, it is assumed to be Le [15].

SW1 SW2: SW1 and SW2, collectively known as "Status Bytes," are mandatory fields within the Response APDU (R-APDU) that indicate the status or result of the executed command.

SW1 (Status Byte 1) and SW2 (Status Byte 2) are both 1-byte fields. They provide information about the execution status, errors, or specific conditions encountered during the processing of the command by the card application as illustrated in Figure 15.

The values of SW1 and SW2 are protocol-specific and defined by the application or card standard being followed. Different combinations of SW1 and SW2 values have different meanings and implications, allowing the terminal application to interpret the response from the card application correctly as illustrated in Figure 16.

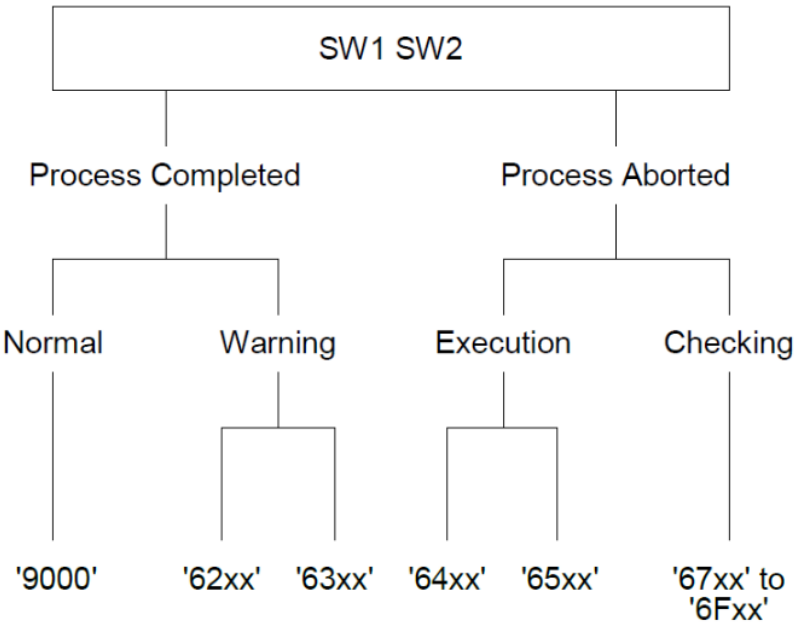


Figure 15:Structural Scheme of Status Bytes *[22]*

SW1	SW2	Meaning
		Normal processing
'90'	'00'	Process completed (any other value for SW2 is RFU)
		Warning processing
'62'	'83'	State of non-volatile memory unchanged; selected file invalidated
'63'	'00'	State of non-volatile memory changed; authentication failed
'63'	'Cx'	State of non-volatile memory changed; counter provided by 'x' (from 0-15)
		Checking errors
'69'	'83'	Command not allowed; authentication method blocked
'69'	'84'	Command not allowed; referenced data invalidated
'69'	'85'	Command not allowed; conditions of use not satisfied
'6A'	'81'	Wrong parameter(s) P1 P2; function not supported
'6A'	'82'	Wrong parameter(s) P1 P2; file not found
'6A'	'83'	Wrong parameter(s) P1 P2; record not found
'6A'	'88'	Referenced data (data objects) not found

Figure 16: Coding of Status Bytes SW1 SW2 *[22]*

2.6.6 TERMINAL RISK MANAGEMENT

During the initial development and deployment of EMV technology, electronic terminals did not always have a continuous connection to the network to authorize each transaction online. However, with advancements in technology, modern terminal manufacturers now typically offer various wireless connectivity options as a standard feature. As a result, offline authorization has become less common and is now considered a niche feature for specific applications.

Offline authorization refers to the capability of a card to make transaction decisions without relying on constant communication with the issuer. Instead, the card follows predefined rules, which are typically defined by thresholds and counters. Based on these rules, the card can independently decide to authorize the transaction on the spot without consulting the issuer.

In certain situations, the card may instruct the terminal to go online and request authorization from the issuer directly. Alternatively, the card may attempt an online authorization, and if a network connection is not available, it can approve the transaction offline.

The terminal performs risk management checks to assess the transaction's risk level. There are three major types of risk checks. They are the floor limit checking when the terminal goes online for transactions over a particular accumulated limit, random transaction selection when the terminal relies on a random number to authorize a particular transaction online, and velocity checking when the terminal checks a soft and a hard limit for number of offline transactions that are allowed since an online card authorization [15].

2.6.7 GENERATE CRYPTOGRAM

Up to two cryptograms can be requested from the card by the terminal through the issuance of Generate AC (GAC) commands. The EMV transaction can either be declined or fully authorized based on the outcome of both the first and second GAC. Here is a breakdown of the steps involved:

1. First GAC: The terminal sends a Generate AC command to the card, requesting the first cryptogram. The card performs various checks and processes the transaction data to generate a cryptogram. The first GAC may also return an authorization request cryptogram (ARQC), which is a type of cryptogram that needs to be sent to the issuer for approval.
2. Authorization Request Cryptogram (ARQC): If the first GAC returns an ARQC, the terminal is responsible for transmitting it to the issuer. The issuer validates the ARQC, verifies the transaction details, and decides whether to approve or decline the transaction. The issuer's response is sent back to the terminal.
3. Second GAC: Once the issuer's response is received, the terminal completes the transaction by issuing the second Generate AC command (second GAC) to the card. This command is used to obtain either a full authorization cryptogram or a full decline cryptogram, depending on the outcome of the issuer's response.
4. Full Authorization or Decline Cryptogram: The card processes the transaction data again based on the issuer's response and generates a cryptogram indicating either full authorization or full decline of the transaction.

5. Transmission to Payment Scheme: The obtained cryptogram either full authorization or full decline is transmitted from the terminal to the payment scheme for clearing. The payment scheme acts as an intermediary, facilitating the transfer of funds between the cardholder's bank and the merchant's bank.

2.7 TRANSACTION COMPLETION

The EMV transaction reaches its conclusion with the second call to the GAC command and the following generation of TC or AAC cryptograms. The TC serves as additional evidence to the issuer, confirming the successful completion of the entire transaction process. It is transmitted to the issuer to further validate the legitimacy of the transaction.

2.8 ATTACKS ON EMV CHIP AND PIN CARDS

EMV chip and PIN cards use advanced security features to protect against fraud, but they are not entirely immune to attacks. Here are some of the different types of attacks on EMV chip and PIN cards:

- **Skimming**: This is the most common attack on EMV chip and PIN cards. Skimming involves stealing card data by placing a device over the card reader, which captures the data from the card's magnetic stripe. In addition to physical skimming devices, thieves can also use technology such as Bluetooth to remotely collect credit card data from a skimming device without having to retrieve it physically. This type of skimming is known as blue skimming and can be carried out by an attacker in close proximity to the victim. Criminals can then use this data to create a clone of the card or make fraudulent purchases. Figure 17 displays an example of an ATM skimmer in action, taken from a research study titled Chip & PIN is definitely broken: Credit Card skimming and PIN harvesting in an EMV world. This research was conducted by Inverse Path, a specialized company in hardware security [23].



Figure 17: ATM skimmer in action

- **Shimming**: Shimming is a newer type of attack that targets the EMV chip and PIN cards. Unlike skimming, which targets the magnetic stripe, shimming targets the chip itself. The attack involves the use of a small device that is inserted into the card reader or the ATM machine, which then intercepts the communication between the chip and the terminal. The device used for shimming is a thin, flexible circuit board that has been designed to fit inside the card reader. Figures (18,19) displays an example of an ATM skimmer in action, taken from a research study titled Chip & PIN is definitely broken: Credit Card skimming and PIN harvesting in an EMV world. This research was conducted by Inverse Path, a specialized company in hardware security [23].

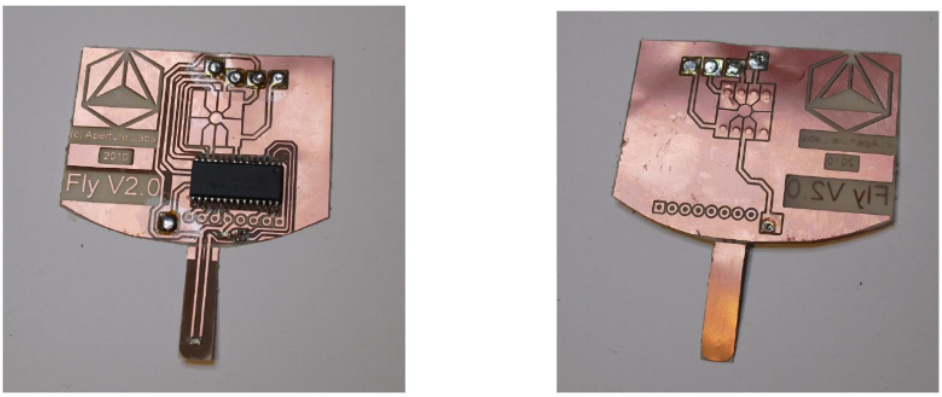


Figure 18: Card Shimming

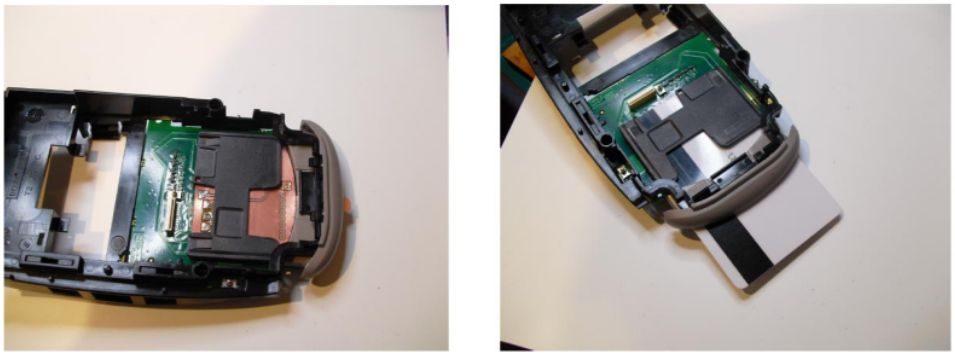


Figure 19: Card reader Shimming

- **Card-Not-Present fraud**: This type of fraud occurs when a criminal uses stolen card data to make purchases online or over the phone, where the card is not physically present. CNP fraud is more common with EMV chip and PIN cards than with magnetic stripe cards, as the chip is more difficult to clone.
- **Malware attacks**: Malware attacks involve installing malicious software on the payment terminal or ATM, which can intercept card data as it is entered or transmitted. This type of attack is more difficult to carry out, as it requires physical access to the payment terminal or ATM.
- **PIN interception**: This type of attack involves intercepting the cardholder's PIN as it is entered into the payment terminal or ATM. Criminals can do this by installing a small camera near the keypad or by using a skimming device that also captures the PIN.
- **Relay attacks:** Relay attacks involve intercepting the communication between the card and the payment terminal and relaying it to a second device, which can be located in a different location. This allows criminals to make fraudulent purchases without physically having the card or the PIN.
- **Man-in-the-Middle:** Is a type of attack where an attacker intercepts the communication between two parties to steal or modify information. In the context of EMV chip and PIN cards, a MITM attack can occur when an attacker intercepts the communication between the card's chip and the payment terminal, and then alters the data in real-time. To carry out a MITM attack on an EMV chip and PIN card, the attacker needs to be in a position to intercept the communication between the card and the payment terminal. This can be done by placing a device between the card and the terminal or by hacking into the payment network. Once the attacker has intercepted the communication, they can modify the data in real-time to carry out a fraudulent transaction.

3 RESEARCH METHODOLOGY

3.1 INTRODUCTION

This section offers a concise overview of the research methodology steps that will be employed in the proposed project. Familiarity with various methods and techniques will aid the author in successfully completing the project and attaining the desired objectives. The chapter outlines key aspects such as research review, philosophy, approach, strategies, choices, time horizon, data collection, research design, and conclusion, providing a brief description of each method within its respective section. The research methodology serves the purpose of providing a systematic approach that enables the researcher to address the problems existing within the chosen field of research. By comprehending these problems based on the research findings, the project can be executed with a well-defined structure, increasing the chances of success and goal achievement.

3.2 RESEARCH OVERVIEW

In this section, the author presents a comprehensive guide to facilitate the successful completion of the project. The chapter consists of several essential sections that are directly related to the project, allowing for the development of a robust research methodology [24]. Figure 20 showcases various methodologies that can be employed, and this chapter serves the purpose of documenting the selected method and justifying its appropriateness for this specific project.

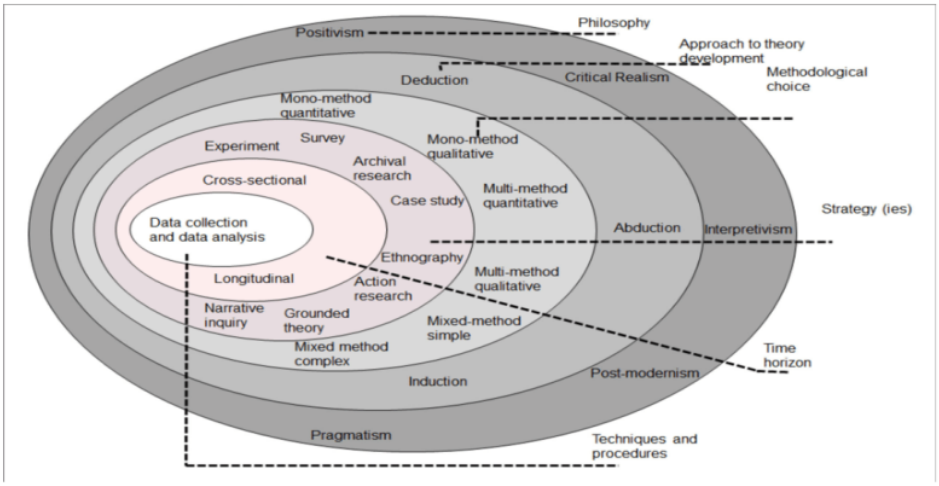


Figure 20: Research Onion *[24]*

3.2.1 RESEARCH PHILOSOPHY

The term research philosophy refers to a system of beliefs and assumptions about the development of knowledge. Although this sounds rather profound, it is precisely what you are doing when embarking on research: developing knowledge in a particular field. The knowledge development you are embarking upon may not be as dramatic as a new theory of human motivation, but even addressing a specific problem in a particular organisation you are, nonetheless, developing new knowledge [25].The choice of research philosophy depends on the nature of the research topic, research questions, and the researcher's epistemological and ontological assumptions. It sets the foundation for selecting appropriate research methods, data collection techniques, and data analysis approaches.

The research philosophy chosen for the project is positivism. Positivism is a suitable research philosophy when the research is based on experiments and aims to rely on scientific evidence. Experiments are considered a form of scientific evidence, and by adopting a positivist approach, the researcher aims to conduct rigorous experiments to gather empirical data and establish causal relationships.

Positivism emphasizes objectivity, systematic observation, and the use of quantitative methods to collect and analyze data. It seeks to identify general laws or patterns that can be applied universally. By employing a positivist research philosophy, the project aims to generate objective and replicable findings that contribute to the body of scientific knowledge.

3.2.2 RESEARCH APPROACH

Author chooses to use both deductive and inductive research approaches; Deductive research involves the formulation of a hypothesis or theory based on existing knowledge and then testing it through the collection and analysis of data. In the case of investigating EMV chip and PIN card security, a deductive approach could be used to test specific hypotheses or theories related to the effectiveness of the technology, such as whether it provides enhanced protection against certain types of fraud or unauthorized access.

Inductive research, on the other hand, involves gathering and analyzing data to identify patterns, trends, or relationships that can then be used to generate hypotheses or develop theories. In the context of EMV chip and PIN card security, an inductive approach could involve analyzing real-world data, such as fraud statistics or case studies, to identify common vulnerabilities, emerging threats, or areas where the security system may be lacking.

It's worth noting that research is often a combination of deductive and inductive approaches. Researchers may start with a deductive approach to test specific hypotheses, but as they collect and analyze data, they may also observe new patterns or make unexpected discoveries, leading to inductive reasoning and the generation of new hypotheses.

Ultimately, the choice between a deductive or inductive approach, or a combination of both, would depend on the research questions, available data, and the goals of the investigation into EMV chip and PIN card security [26].

3.2.3 METHODOLOGICAL CHOICES

The author's choice of methodology is vital for comprehending research findings and making informed decisions. In this study, a mono method approach has been selected, specifically tailored to align with the proposed project. This chosen method incorporates both qualitative and quantitative techniques, each serving a distinct purpose. Qualitative methods are employed to capture textual data, while quantitative methods focus on numerical data collection. Quantitative approaches primarily serve the purpose of hypothesis verification, whereas qualitative methods aim to understand the characteristics of the data samples. Since the project involves conducting an experiment on EMV chip and PIN cards, it requires the collection of both numerical and textual data. As a result, the author has chosen to adopt a mono method approach that combines both qualitative and quantitative methods.

3.2.4 RESEARCH STRATEGIES

A research strategy refers to the overall plan or approach adopted by researchers to conduct their study and achieve their research objectives. It encompasses the key steps, methods, and techniques utilized throughout the research process, and the choice of strategy is influenced by factors such as the research question, available resources, and desired outcomes. In the context of investigating EMV chip and PIN card security, the experimental strategy has been determined as the most suitable for the project.

The author intends to employ a combination of hardware and software for conducting the experiments. The hardware component will involve the use of a desktop computer or a laptop, along with specific devices such as the SIMtrace2 device [27], a card machine, bank card, smart card reader, and smart card. Regarding the software aspect, the setup will include an internet

connection, Linux Ubuntu operating system, SIMtrace 2 firmware, Wireshark packet sniffer, and other software capable of sniffing network traffic. Through the utilization of these tools, the author aims to conduct MITM attack experiments to identify vulnerabilities in EMV chip and PIN card security.

3.2.5 TIME HORIZONS

The time horizon refers to the defined time frame within which the project is intended to be completed. In this study, a cross-sectional method is employed to establish a sample of extensive data discovered during the research, forming a solid foundation. The cross-sectional approach allows the researcher to investigate a specific phenomenon at a particular moment in time. By focusing on the present situation, cross-sectional or short-term study involving collection of data at a specific point of time will be suitable for the research [26]. The time horizon serves as a plan that ensures each task is completed within the designated time frame, thereby facilitating the successful completion of the project as a whole.

3.2.6 DATA COLLECTION METHODS

Data collection is the systematic process of gathering relevant information or data to address research questions, test hypotheses, or achieve research objectives. It involves the structured collection, recording, and organization of data to establish a basis for analysis and interpretation. The selection of appropriate data collection methods relies on factors such as the nature of the research, research questions, available resources, and ethical considerations.

In this section, a diverse range of methods and techniques will be employed to collect data and investigate the security of EMV chip and PIN cards. The primary goal is to provide recommendations for future enhancements. During the research phase, the author will gather data from reputable sources, including IEEE, eBooks, scientific journals, articles, white papers, and other pertinent fields of study. The collected data will then be utilized to conduct specific experiments focused on EMV cards, with the aim of identifying security risks. Subsequently, the author will meticulously analyze the findings and strive to offer recommendations for future improvements that effectively address vulnerabilities, threats, and attacks associated with EMV chip and PIN cards.

Table 1 below illustrates the correlation between the various chapters of the project and the research onion framework, highlighting how each chapter aligns with different components of the research process. The purpose of the table is to identify and demonstrate the interconnectedness of the project's chapters with the stages of the research onion model.

Component	Research Approach
Chapter 1 – Introduction	**Deductive** reasoning is utilized in this chapter, with the hypothesis positing that EMV cards are insecure based on prior instances of exploitation.
Chapter 2 - Research & review of literature	**Qualitative Research** understanding the EMV protocol and security threats and weaknesses. **Action Research** Gaining a comprehensive understanding of the findings discovered during the research.
Chapter 3 - Research Methodology	**Action Research** Gaining a comprehensive understanding of the findings discovered during the research.
Chapter 4 - Design & Implementation	**Action Research** Gaining a comprehensive understanding of the findings discovered during the research. **Experiment or Qualitative** Conducting an experiment.
Chapter 5 - Testing and Evaluation	**Qualitative** Collecting the outcomes/data. **Action Research** Gaining a comprehensive understanding of the findings discovered during the research.
Chapter 6 - Conclusion	**Action Research** Gaining a comprehensive understanding of the findings discovered during the research.
Chapter 7 - Reflections	**Qualitative** The extent of reflection depends on the specific findings and results obtained from the experiment
Chapter 8 – References	**Qualitative** Offers up-to-date and credible references.

Table 1: Data Collection Methods

3.3 RESEARCH DELIVERABLES

This research study offers several deliverables as it is supported by an extensive literature review that incorporates previous study findings. These sources serve as the foundation for conducting testing and experiments. By critically evaluating the outcomes of the experiments, the author establishes connections between the research findings and the experimental results. The proposed experiments specifically focus on EMV chip and pin card security and aim to identify vulnerabilities, threats, and attacks that users may encounter while using these cards. The ultimate aim of the study is to offer effective solutions that protect cardholder information from compromise by attackers.

3.4 CONCLUSION

In conclusion, this research project embraced a comprehensive research onion approach, integrating various components such as research philosophy, approach, strategies, choices, time horizons, and data collection methods. The adoption of a positivist research philosophy facilitated the measurement of a substantial data sample during the experimental testing phase. By employing both inductive and deductive approaches, the research was able to pose precise questions, seek answers, and establish hypotheses.

The chosen strategies enabled the utilization of a hardware and software combination, resulting in a comprehensive experimental setup and successful project completion. The incorporation of both qualitative and quantitative methods offered a comprehensive understanding of data sample characteristics and allowed for the verification of experiment results against hypotheses.

Consideration of the time horizon ensured adherence to the project timeline and successful completion within the specified timeframe. This chapter played a pivotal role in the research process, employing a variety of methods and techniques to achieve project objectives and desired outcomes.

The research deliverables include a comprehensive research report, and recommendations for future enhancements in EMV card security. These outputs serve as valuable resources for stakeholders in the financial industry, such as card issuers, financial institutions, and regulatory

bodies, empowering them to develop robust strategies and implement effective measures that protects cardholders from potential threats.

4 DESIGN & IMPLEMENTATION

4.1 INTRODUCTION

This section provided a concise overview of the entire process, covering experiment design and implementation. The primary objective of this chapter was to furnish essential details required to execute the experiment seamlessly. The experiment design and implementation section identified potential issues that might have arisen during the experiment and proposed corresponding solutions. The implementation section offered a concise description of the particular tools and software carefully chosen by the author to ensure the project's success. Each tool and software utilized was explicitly listed within this section. Lastly, the author concluded by briefly discussing any challenges encountered during the design and implementation phases, with the aim of enhancing clarity and understanding for the readers.

4.2 PROBLEM FORMULATION

The main objective of the Investigating EMV Chip and PIN Card Security project was to evaluate the security of widely used EMV chip and PIN cards during payment transactions. The initial phase of the investigation involved identifying the necessary tools for executing a MITM attack and capturing real-time responses. These tools included a mix of hardware and software components. Before conducting the experiment on actual bank cards, a preliminary test was conducted on a similar device, like a phone, to ensure the effectiveness of the MITM attack. This pre-test verified that real-time responses could be intercepted and accurately analyzed, particularly since the communication protocol used by the SIM card and chip in the EMV card was the same.

The final stage of the investigation entailed carrying out the experiment on real bank cards, where the communication was intercepted and the intercepted packets were analyzed. This process aimed to gain deeper insights into the card details, potentially revealing vulnerabilities or weaknesses. This analysis was critical to ensure that sensitive card information was not susceptible to falling into the wrong hands and to propose suitable security enhancements for safeguarding against potential threats.

4.2.1 TOOLS AND PREREQUISITES

To accomplish the project successfully and meet the desired goals, the author needed to utilize diverse resources and plan accordingly. The design and implementation of the project heavily relied on a combination of both hardware and software components.

Hardware

- **Laptop or desktop**: The laptop will serve as the main computing device for executing tasks, running necessary software, and conducting experiments. It will play a crucial role in achieving the project's objectives and meeting the target goals.
- **SIMtrace2 board** [27]**:** Is a specialized hardware device designed to facilitate interaction between a mobile device's SIM card and the mobile network. It is available for purchase on the sysmocom workshop website at a price of €101.15. With its capabilities, researchers, developers, and security professionals can effectively monitor, capture, and analyze communication exchanges between the SIM card and the mobile network. SIMtrace2 finds widespread application in forensic analysis, vulnerability assessments, and mobile communication security testing. Its usage allows for a deeper comprehension of the intricate interactions between SIM cards and mobile networks, rendering it an invaluable tool in various telecommunications and security-related domains. Figure 21 shows the SIMtrace2 board.

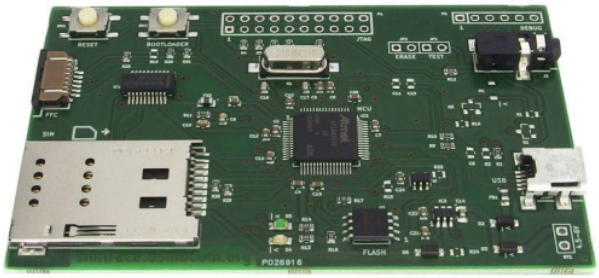


Figure 21:SIMtrace2 Hardware Kit

- **Flexible PCB SIM adapters** [27]**:** Are specialized printed circuit board (PCB) devices used to interface and adapt different types of SIM cards with various devices or systems. These adapters are designed to convert the physical format of a SIM card, allowing it to be used in a different SIM card slot or reader. They provide a bridge between the unique form factors of SIM cards and the target device's specific SIM card slot. For the experiment the author needs a set of four flexible PCB SIM adapters for 2FF SIM card size, one in each orientation. Which can be bought for €45.00 on the sysmocom workshop website [27]. Figure 22 shows PCB SIM adapter.

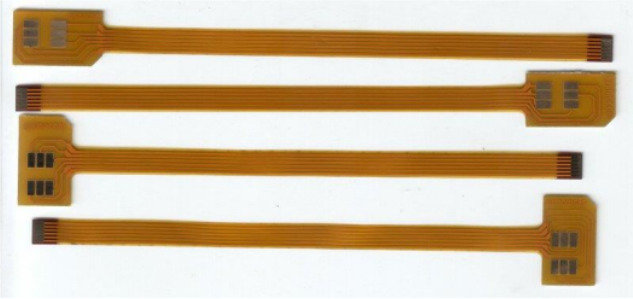


Figure 22:SIMtrace microSIM (3FF) FPC Cable

- **Smart Card FPC Cable:** It refers to a Smart Card with a Flexible Printed Circuit (FPC) attached to it. which can be bought for €21.42 on the sysmocom workshop website [27]. This combination of a Smart Card with a Flexible Printed Circuit enables seamless connectivity and communication between the card's chip and external components, making it a valuable solution for various secure and portable electronic systems. Figure 23 shows the smart card and the FPC cable.

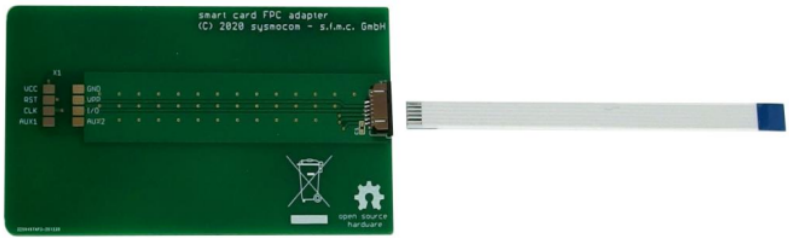


Figure 23:SIMtrace Smart Card FPC

- **SIM card to FPC adapter (card reader):** The SIM card to FPC adapter simplifies the integration of SIM card functionality into systems that use FPC connectors, offering a flexible and convenient solution for SIM card compatibility in various electronic products. which can be bought for €23.80 on the sysmocom workshop website [27]. Figure 24 shows the smart card adapter.



Figure 24:SIM card to FPC adapter

- **Card machine:** A card machine, also referred to as a card terminal, POS terminal, or credit card reader, served as a tool for processing electronic payments via debit cards, credit cards, or other payment cards. These devices were commonly employed by businesses and merchants to enable seamless cashless transactions. For this experiment, the author obtained a card machine from a friend who owned a small business. This card machine was instrumental in conducting the experiment, as it allowed the author to securely and conveniently process electronic payments during their research or project.



Figure 25: Card Machine

Software

- **Linux operating system**
- **Osmocom SIMtrace 2:** Is a software, firmware, and hardware system for passively tracing SIM-ME communication between the SIM card and the mobile phone, and remote SIM operation. While it was designed for SIM-ME communication, it supports all ISO 7816 smart-cards using the T=0 protocol (the most common case) [27].
- **Wireshark packet sniffer**: Is a widely used open-source network protocol analyzer. It allows users to capture, inspect, and analyze network traffic in real-time or from saved files.

4.3 PROJECT DESIGN

4.3.1 PRE-EXPERIMENT DESIGN

As mentioned earlier, both the SIM card and EMV chip employed the same communication protocol. To verify the seamless operation of all essential hardware and software components for the main experiment, the author performed this test. Figure 26 shows the design of the pre-test

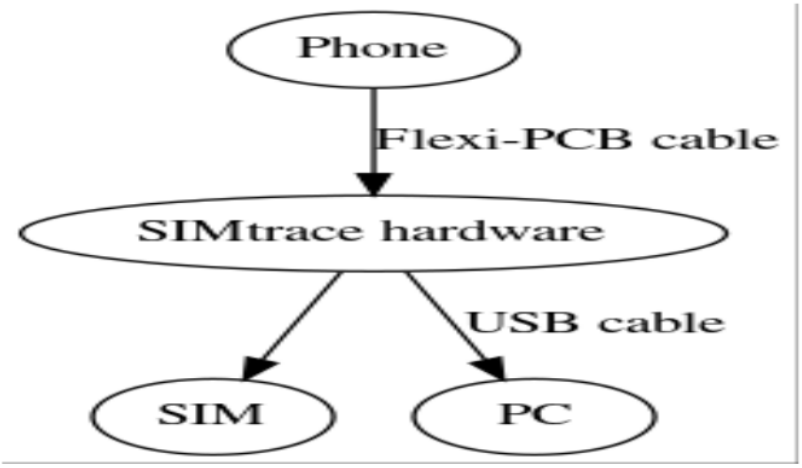


Figure 26: pre-test design

4.3.2 EXPERIMENT DESIGN

The author's experiment design comprises a laptop, Simtrace2 device, card reader, fake card, and a card machine. This setup shown in Figure 27 allows the author to conduct the experiment and gain a better understanding of how the Simtrace2 device functions as a communication diversion. The Simtrace2 device diverts all communication to the laptop, enabling the interception of data through packet sniffing software such as Wireshark and Simtrace2-sniff.

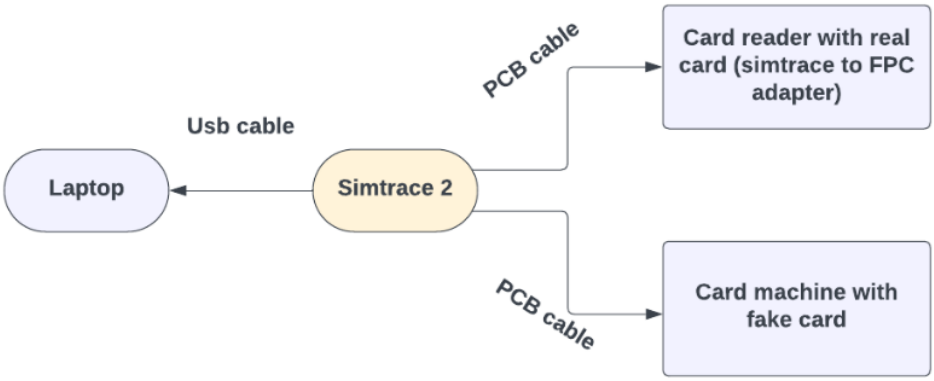


Figure 27: Experiment Design

4.4 IMPLEMENTATION

4.4.1 UBUNTU 22.04.2 LTS

Ubuntu 22.04.2 LTS [28], codenamed Jammy Jellyfish, is the most recent stable release of the renowned Linux-based operating system. This version includes numerous enhancements, bug fixes, and security updates to guarantee a trustworthy and safe user experience. Being designated as LTS (Long-Term Support) means it will receive official updates and support for an extended duration, making it a reliable option for both desktop and server setups. The author chose this Linux distribution for their experiment and has successfully installed it on their laptop, as depicted in the Figure 28.



Figure 28: Ubuntu 22.04.2 LTS

4.4.2 OSMOCOM SIMTRACE 2

Osmocom SIMtrace 2 [27] is a software, firmware and hardware system for passively tracing SIM-ME communication between the SIM card and the mobile phone, and remote SIM operation. While it was designed for SIM-ME communication, it supports all ISO 7816 smart-cards using the T=0 protocol (the most common case).

This software has been previously installed using the following commands shown in Figures (29,30,31,32,33) .

```
sudo apt-get install simtrace2-utils.
```

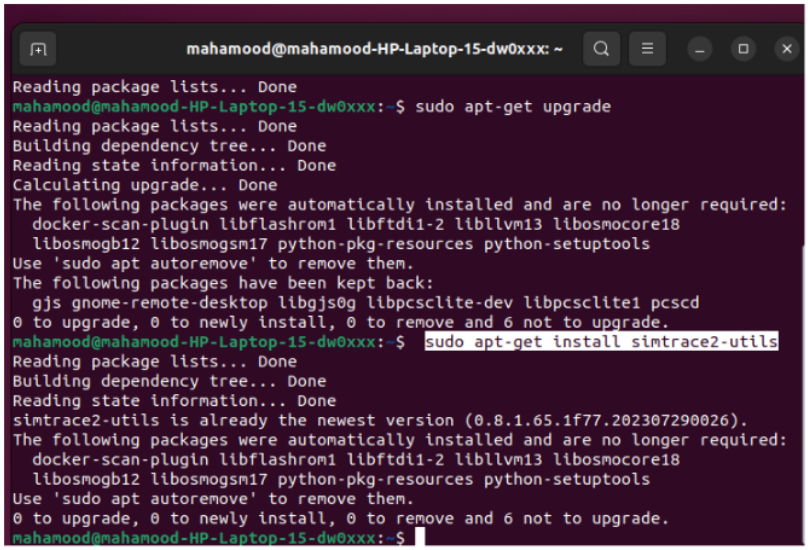


Figure 29: Osmocom SIMtrace 2

sudo apt-get install libusb-1.0-0-dev libosmocore-dev libpcsclite-dev.

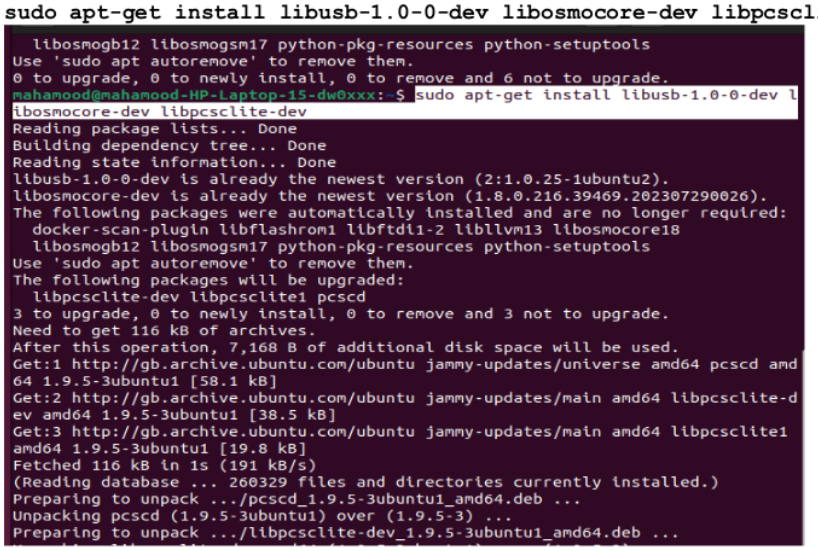


Figure 30: Osmocom SIMtrace 2

```
git clone https://gitea.osmocom.org/sim-card/simtrace2.git
cd simtrace2/host/
autoreconf -fi
./configure
Make
```

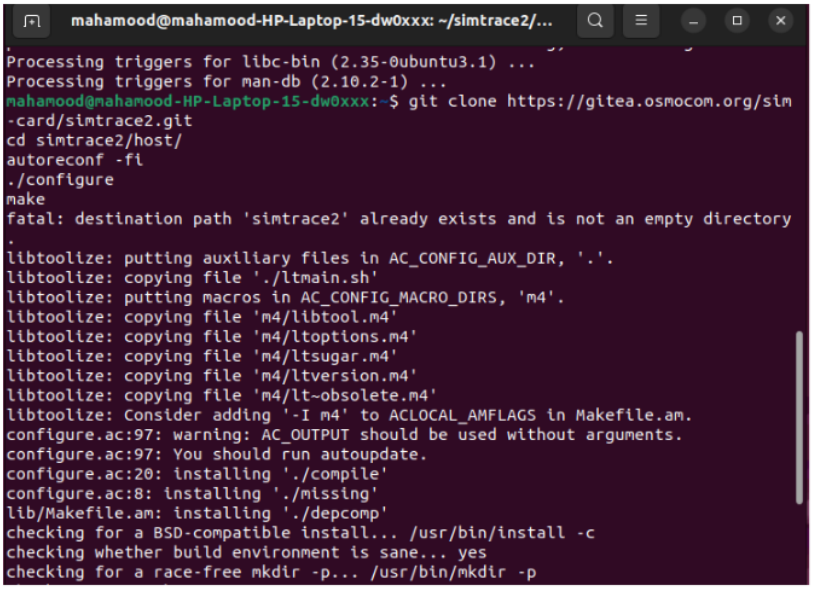


Figure 31: Osmocom SIMtrace 2

```
sudo adduser $USERNAME plugdev
sudo wget -O /etc/udev/rules.d/99-simtrace2.rules
https://gitea.osmocom.org/sim-
card/simtrace2/raw/branch/master/host/contrib/99-simtrace2.rules
sudo udevadm control --reload-rules
sudo udevadm trigger
```

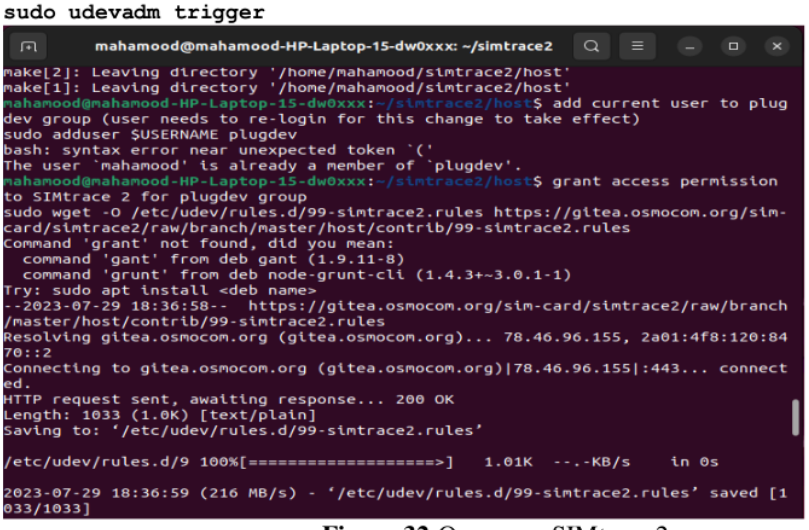


Figure 32: Osmocom SIMtrace 2

```
To check if it Was installed correctly.
cd simtrace2
simtrace2-list
```

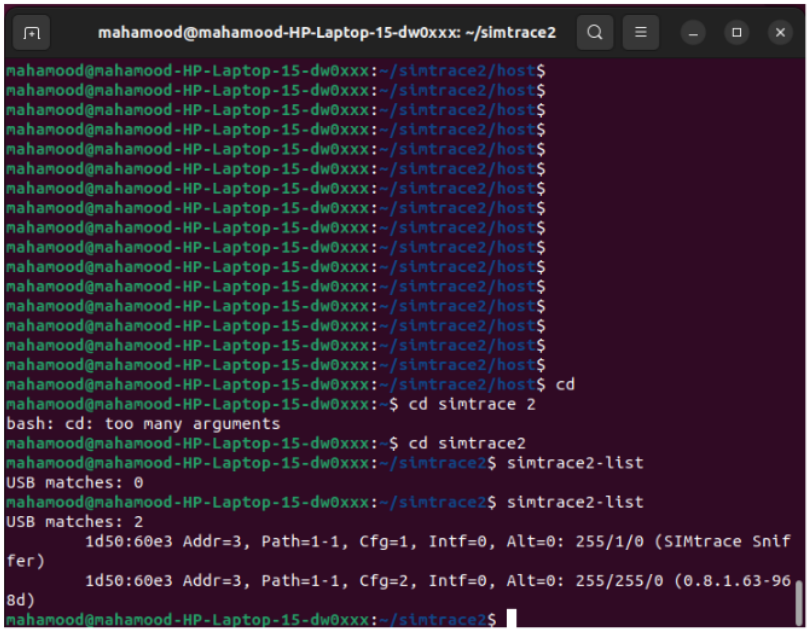


Figure 33: Osmocom SIMtrace 2

4.4.3 WIRESHARK PACKET SNIFFER

Wireshark [29] is a popular and powerful open-source packet sniffer and network protocol analyzer. It allows users to capture and inspect network traffic in real-time, providing a detailed view of the data packets flowing through a network. With Wireshark, users can analyze and interpret various network protocols, troubleshoot network issues, and gain valuable insights into network behaviour and performance. It is widely used by network administrators, security professionals, and researchers to monitor and analyze network traffic for diagnostic, security, and research purposes. The author utilized Wireshark to analyze the intercepted packets. This software has been previously installed using the following commands shown in Figure (34,35).

```
sudo apt install wireshark
```

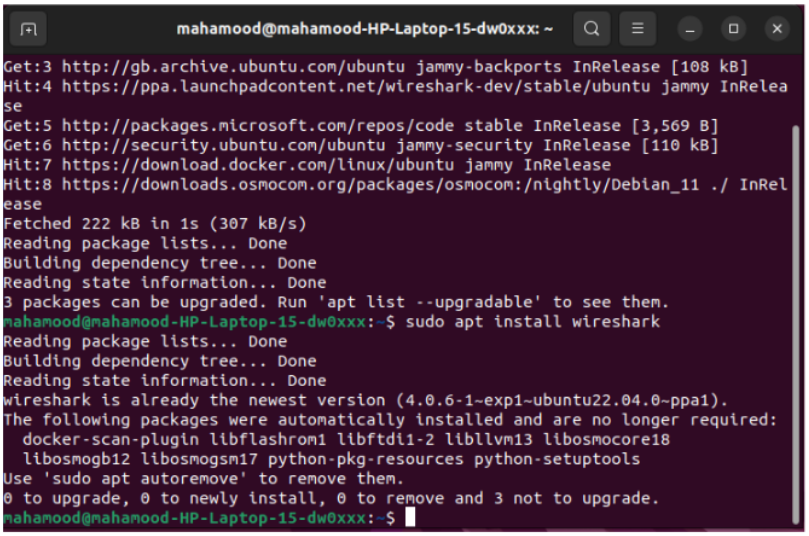


Figure 34: Wireshark

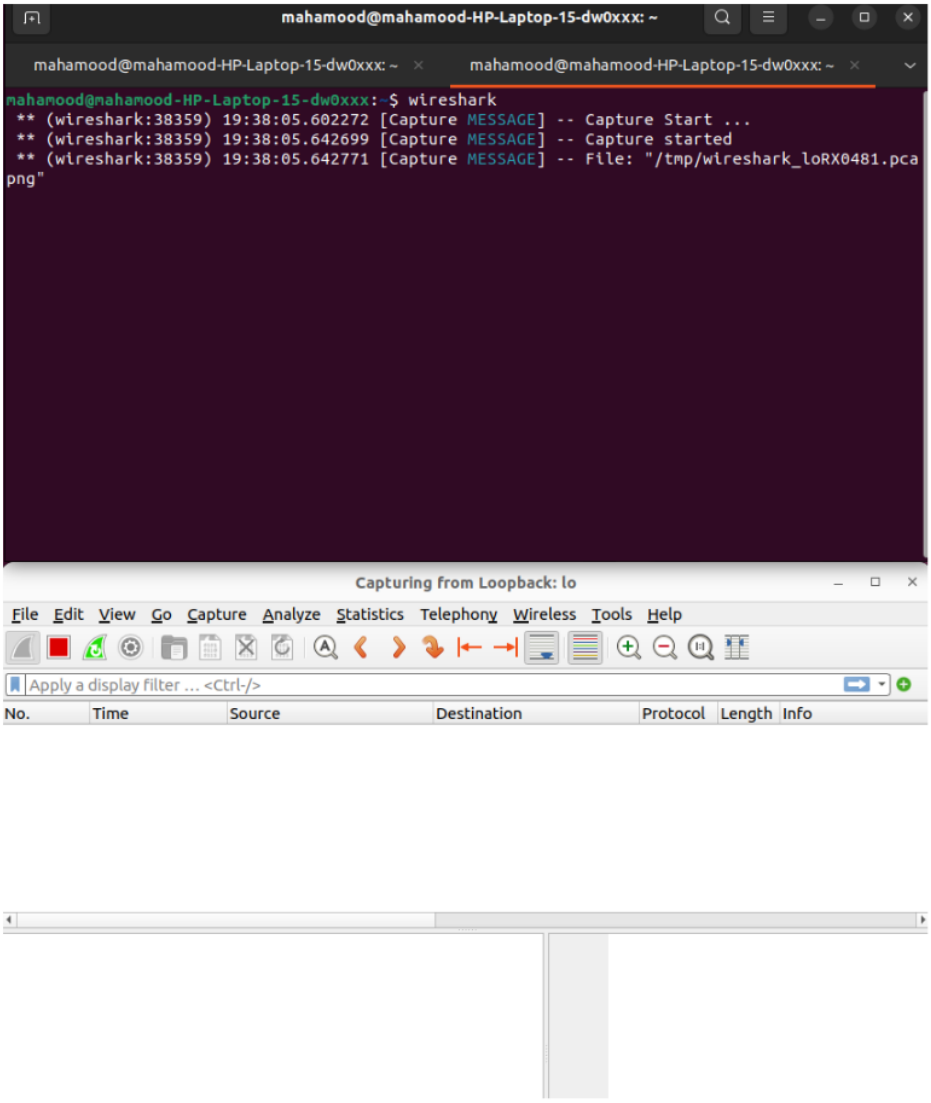


Figure 35: Wireshark

4.4.4 HARDWARE IMPLEMENTATION

In the previous section, the author provided detailed instructions on how to install and utilize each component of the hardware and software. In this section, the author showcased actual photographs of the fully implemented hardware setup for both the experiment pre-test and the experiment test showed in Figure (36,37).

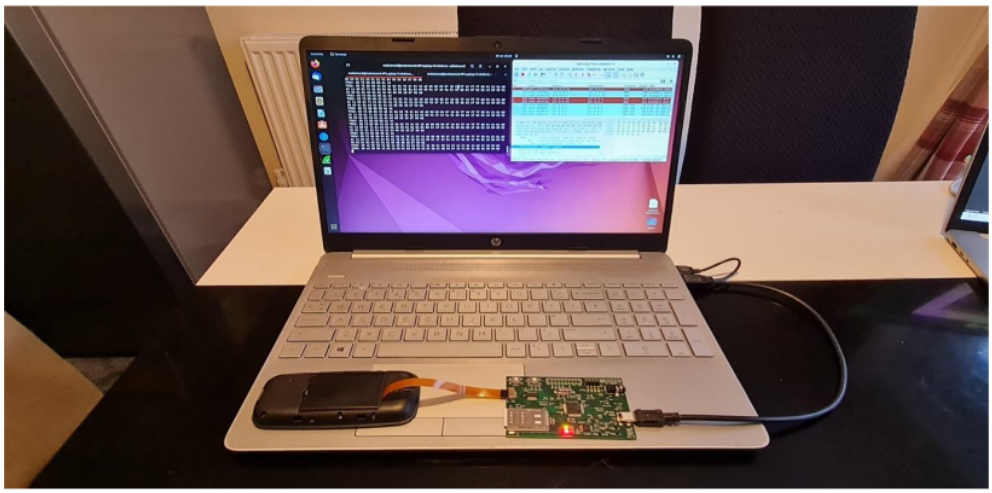


Figure 36: Experiment pre-test



Figure 37: Experiment test

4.4.5 SUMMARY

This chapter is of utmost importance in this research, as it allows the author to conduct an in-depth exploration of the EMV protocol. However, the author encountered certain technical challenges while installing the required packages for the simtrace2 device.

Despite these hurdles, the hardware and software tools introduced in this chapter proved instrumental in facilitating the successful execution of the experiments. The results and findings obtained from these experiments were comprehensively discussed in chapter 5.

5 TESTING AND EVALUATION

5.1 INTRODUCTION

In this chapter, the primary emphasis was on testing and evaluating the results obtained from the conducted experiments in the investigation of EMV Chip and PIN card security. The testing process involved a detailed analysis of the data captured during the experiment, specifically focusing on the APDU for the request and response commands. The author aimed to delve into these APDUs to identify the card details that could be obtained from them. By thoroughly examining these APDUs, valuable insights into the functioning of the EMV Chip and PIN card system were gained, contributing to a deeper understanding of its security aspects.

5.2 TESTING RESULTS DISCUSSION

5.2.1 TEST ONE, PRE-EXPERIMENT TEST

The objective of this test is to ensure the smooth operation of all critical hardware and software components required for the main experiment, enabling the author to confirm that all the equipment is functioning as expected.

The interception of APDUs can be observed in both Wireshark and simtrace2-sniff, as depicted in the Figure 38, Figure 39. It confirms that the ISO/IEC 7816-4 APDU has been captured during the process.

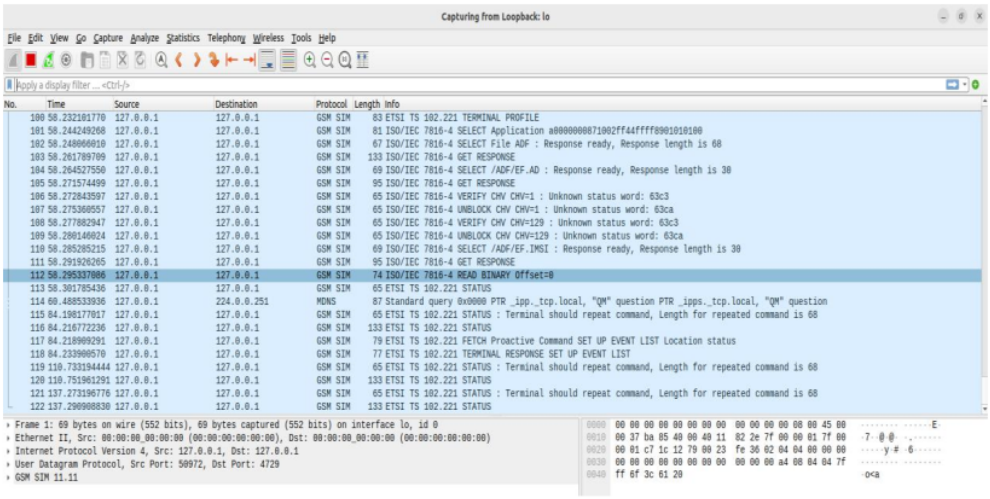


Figure 38: Wireshark simcard sniffing.

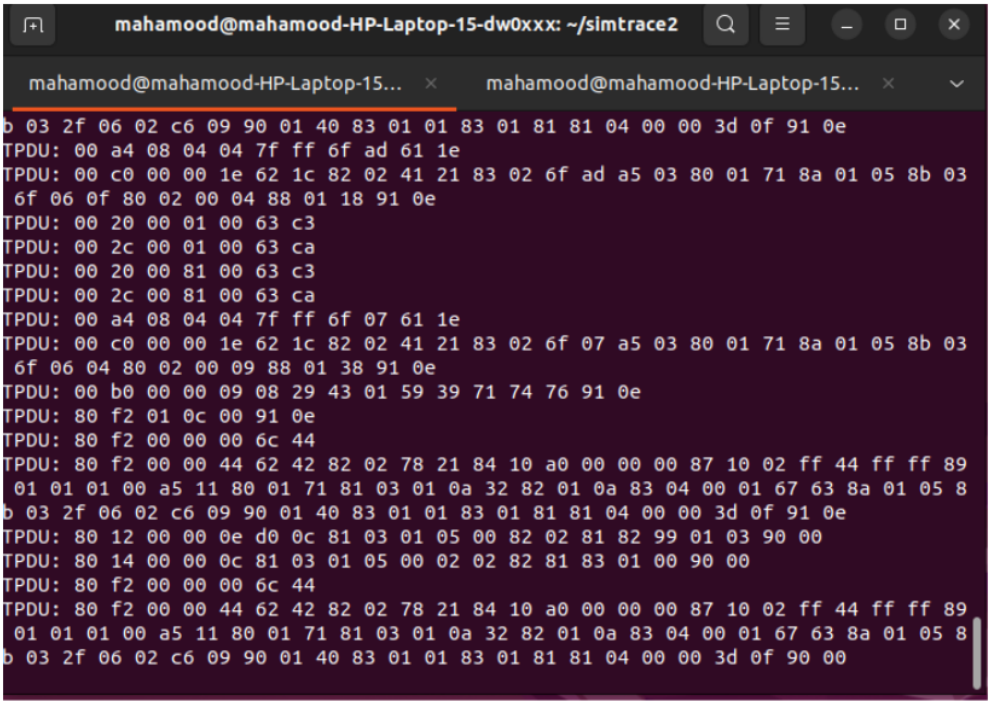


Figure 39: Simtrace2 sniffing.

5.2.2 TEST TWO, SNIFFING BANK DEBIT CARDS DURING REALTIME TRANSACTION

This test demonstrated the card's interaction with the terminal and the bank, essentially detailing the entire payment procedure in real time. The author clarified each APDU, both requests and responses, and explicated their significance in relation to the EMV protocol and the payment process. The author carried out MITM attack on multiple bank cards, Figure 40 and Figure 41 showed the test results using both the Wireshark and Linux terminal.

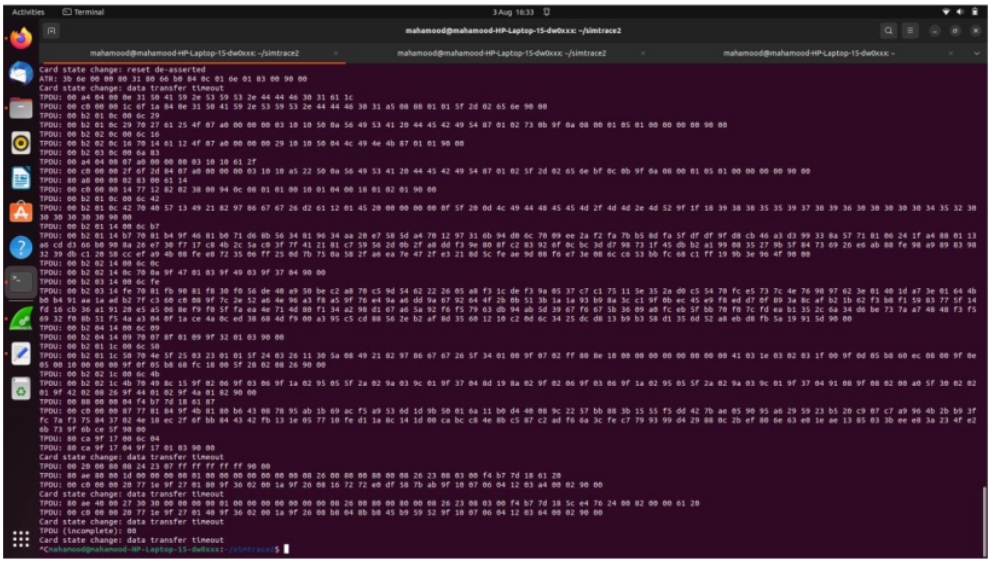


Figure 40: Bank card sniffing

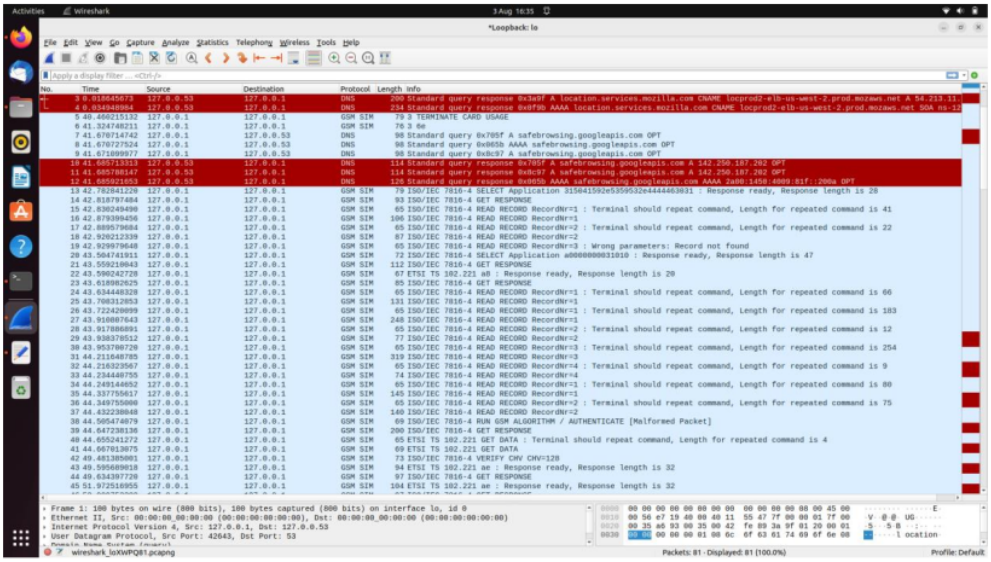


Figure 41: Bank card sniffing Wireshark.

5.2.2.1 MITM ATTACK ON CARD 1 (HALIFAX VISA CARD)

ATR: 3b 6e 00 00 80 31 80 66 b0 84 0c 01 6e 01 83 00 90 00

The ATR is a message that was output by a contact smart card upon initial powering, as observed by the author, in response to an RST signal sent from the card machine. Information about the card's properties and communication parameters was noted by the author to be contained within it. The complete details of the ATR can be found in **Appendix 1**.

Request 1 [30]**: TPDU: 00 a4 04 00 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 30 31 61 1c**

The select command for an EMV application directory was identified by the author. This command, sent from the terminal to the card, requests the card to select the EMV Payment System Directory (PSE), which, as observed by the author, is 1PAY.SYS.DDF01. The complete details of the request 1 can be found in **Appendix 1.1**.

Response 1 [30]**: TPDU: 00 c0 00 00 1c 6f 1a 84 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 30 31 a5 08 88 01 01 5f 2d 02 65 6e 90 00**

The get response command, which asked for 28 bytes of data related to a previously sent command (request 1), was identified by the author. The response, as was observed by the author, provided details of the 1PAY.SYS.DDF01 EMV application directory, including its priority and language preference (English). The complete details of the response 1 can be found in **Appendix 1.1**.

Request 2 [30]**: TPDU: 00 b2 01 0c 00 6c 29**

The author identified the read record command, which had requested the card to read a specific record (record 1) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 41 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback. The complete details of the request 2 can be found in **Appendix 1.2**.

Response 2 [30]**: TPDU: 00 b2 01 0c 29 70 27 61 25 4f 07 a0 00 00 00 03 10 10 50 0a 56 49 53 41 20 44 45 42 49 54 87 01 02 73 0b 9f 0a 08 00 01 05 01 00 00 00 00 90 00**

The data returned by the card in response to Request 2 was analyzed by the author and was found to include various information. An application identifier (4f 07 a0 00 00 00 03 10 10) for a specific card application and the application label VISA DEBIT were among the details noted by the author. The complete details of the response 2 can be found in **Appendix 1.2**.

Request 3 [30]**: TPDU: 00 b2 02 0c 00 6c 16**

The author identified the read record command, which had requested the card to read a specific record (record 2) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 22 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback. The complete details of the request 3 can be found in **Appendix 1.3**.

Response 3 [30]**: TPDU: 00 b2 02 0c 16 70 14 61 12 4f 07 a0 00 00 00 29 10 10 50 04 4c 49 4e 4b 87 01 01 90 00**

The data that was returned by the card in response to the command (Request 3) was analyzed by the author and was found to include various details. An application identifier (4f 07 a0 00 00 00 00 29 10 10) for a specific card application and 50 04 4c 49 4e 4b, which the author identified as standing for the application label (LINK), were among the details noted. The end of the TPDU (90 00), as observed by the author, indicated a successful operation or no error. The complete details of the response 3 can be found in **Appendix 1.3**.

Request 4 [30]**:TPDU: 00 b2 03 0c 00 6a 83**

The author identified the read record command, which had requested the card to read a specific record (record 3) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested record did not exist. The complete details of the request 4 can be found in **Appendix 1.4**.

Request 5 [30]**: TPDU: 00 a4 04 00 07 a0 00 00 00 03 10 10 61 2f**

The select command, which was asking the card to select the Visa Debit application, was identified by the author. It was noted by the author that the card responded by indicating that there were 47 more bytes of data available, which could be obtained by sending a get response command. The complete details of the request 5 can be found in **Appendix 1.5**.

Response 5 [30]**: TPDU: 00 c0 00 00 2f 6f 2d 84 07 a0 00 00 00 03 10 10 a5 22 50 0a 56 49 53 41 20 44 45 42 49 54 87 01 02 5f 2d 02 65 6e bf 0c 0b 9f 0a 08 00 01 05 01 00 00 00 00 90 00**

The get response command, which represented the response of a previously issued select command (Request 5), was identified by the author. The author noted that it provided details of the VISA DEBIT application on the card. The complete details of the response 5 can be found in **Appendix 1.5**.

Request 6 [30]**: TPDU: 80 a8 00 00 02 83 00 61 14**

This was identified by the author as a get processing options command that had been issued without any specific processing options requested. The author noted that the response status indicated that there were still 20 additional response bytes available for this command. The complete details of the request 6 can be found in **Appendix 1.6**.

Response 6 [30]**: TPDU: 00 c0 00 00 14 77 12 82 02 38 00 94 0c 08 01 01 00 10 01 04 00 18 01 02 01 90 00**

The get response command, which represented the response of a previously issued get processing options command (Request 6), was identified by the author. It was observed by the author that this command retrieved a response message with details about the card's capabilities and records to be read. The complete details of the response 6 can be found in **Appendix 1.6**.

Request 7 [30]**: TPDU: 00 b2 01 0c 00 6c 4b**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 1) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 75 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback. The complete details of the request 7 can be found in **Appendix 1.7**.

Response 7 [30]**: 00 b2 01 0c 4b 70 49 57 13 44 62 91 93 90 53 65 83 d2 70 52 01 61 10 00 00 00 00 0f 5f 20 16 4c 49 44 48 45 45 4d 2f 4d 41 48 4d 4f 4f 44 20 4d 20 53 2e 4d 52 9f 1f 18 30 37 36 39 32 37 35 30 30 36 37 30 30 30 30 36 31 31 30 30 30 30 30 30 90 00**

The data that was read from the card, as noted by the author, contained track data, the cardholder's name (LIDHEEM/MAHMOOD M S.M), and other discretionary data. It was observed by the author that the READ RECORD command had been successful, and the card had provided the requested data. The complete details of the response 7 can be found in **Appendix 1.7**.

Request 8 [30]**: TPDU: 00 b2 01 14 00 6c b7**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 1) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 183 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback. The complete details of the request 8 can be found in **Appendix 1.8**.

Response 8 [30]**: TPDU: 00 b2 01 14 b7 70 81 b4 9f 46 81 b0 15 aa 57 89 a8 0a f9 c8 85 eb b3 ae 8d 39 b6 77 18 bd ff 54 f3 5d 8f b0 2b 17 02 ab 11 24 d3 7c b3 44 63 af fe ee 4c 74 f9 3b 4b 83 d0 14 b6 a9 83 93 77 2f c8 77 9e 24 bc 28 cc 62 3e d4 53 80 05 f4 ab 87 f7 a3 0a 2c 4a 48 40 e5 3e 70 91 78 47 7d 9a 99 b1 01 fd a2 15 f7 0e 66 4f 50 8f 02 67 1d cb ba e2 a0 70 ad 9d 72 0c d2 97 ba cd fe 92 50 b0 bd de 45 d3 eb bf b5 e4 d9 e1 0e 03 e8 f4 36 44 c6 54 f2 1b 18 16 d2 94 55 80 cf 56 a1 0f cb ff 27 2d e2 e6 85 ab 7e 62 f0 8c 85 e9 e2 c8 3e ef ce e4 a1 fc da dc 6a 20 e2 56 97 e8 cc 90 00**

The command, identified by the author, was used to read the record. It was specifically aimed at retrieving the Public Key Certificate from the Integrated Circuit Card (ICC), as noted by the author. The complete details of the response 8 can be found in **Appendix 1.8**.

Request 9 [30]**: TPDU: 00 b2 02 14 00 6c 0c**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 2) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 12 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback. The complete details of the request 9 can be found in **Appendix 1.9**.

Response 9 [30]**: TPDU: 00 b2 02 14 0c 70 0a 9f 47 01 03 9f 49 03 9f 37 04 90 00**

This was identified as a response message by the author, which initiated the card authentication process, as noted in the author's observations. The complete details of the response 9 can be found in **Appendix 1.9**.

Request 10 [30]**: TPDU: 00 b2 03 14 00 6c fe**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 3) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 254 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback. The complete details of the request 10 can be found in **Appendix 1.10**.

Response 10 [30]**: TPDU: 00 b2 03 14 fe 70 81 fb 90 81 f8 23 be 6a be 97 81 83 35 e6 aa 3a ea cd c0 af cf 50 61 55 4d 6a 17 73 62 56 bb 77 d2 84 6a f8 e5 7b 6d 20 4a a7 a9 11 f6 54 bd 4f 96 91 e7 c8 a3 83 5d d6 b0 f0 9b 9f 22 60 6d a9 11 ef 72 5a 63 9c c5 31 8a b7 be eb 9e 63 36 49 28 7a aa 6e b1 f2 37 80 24 62 b9 d9 6d 2f 6d 2f a5 8a cd 71 90 f0 6c b9 7d f8 c2 bb d0 74 db ef 87 28 70 26 1b db c0 5d 28 17 c1 5b 5a e2 de 74 34 cf b3 b7 7a db 4c 40 18 55 89 6f 78 4c 35 c7 21 fe 42 d9 89 8d 27 7c fc dc 62 7f 63 d5 3a 66 f2 bc 8f 74 2f af c9 b8 ea ea 4e 0c 36 f4 5e 9c 64 1d 24 17 e3 f3 23 7a fb 05 83 45 39 28 01 e5 25 6f 9b 1b 0c 8a a4 20**

70 3e 3e a1 fa 8c f0 93 64 84 6f 83 41 a0 f9 56 01 7c a9 1c 91 c2 20 ae 37 91 75 f9 4a 6d b5 42 8f 93 79 24 86 67 73 89 12 d5 c9 33 ea 4b 20 16 4c d2 0e 0d 0b 90 00

The command, identified by the author, was used to read the record. It was specifically aimed at retrieving the Issuer Public Key Certificate, as noted by the author. The complete details of the response 10 can be found in **Appendix 1.10**.

Request 11 [30]**: TPDU: 00 b2 04 14 00 6c 09**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 4) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 9 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback. The complete details of the request 11 can be found in **Appendix 1.11**.

Response 11 [30]**: 00 b2 04 14 09 70 07 8f 01 09 9f 32 01 03 90 00**

As identified by the author, it represented a Read Record command that retrieved data from the card, including details such as the Certification Authority Public Key Index, which was 9, and the Issuer Public Key Exponent value, which was 3. The complete details of the response 11 can be found in **Appendix 1.11**.

Request 12 [30]**: TPDU: 00 b2 01 1c 00 6c 50**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 1) from the current file (SFI 3). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 80 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback. The complete details of the request 12 can be found in **Appendix 1.12**.

Response 12 [30]**: TPDU: 00 b2 01 1c 50 70 4e 5f 25 03 23 04 01 5f 24 03 27 05 31 5a 08 44 62 91 93 90 53 65 83 5f 34 01 00 9f 07 02 ff 80 8e 10 00 00 00 00 00 00 00 00 41 03 1e 03 02 03 1f 00 9f 0d 05 b8 60 ec 08 00 9f 0e 05 00 10 00 00 00 9f 0f 05 b8 68 fc 18 00 5f 28 02 08 26 90 00**

As observed by the author, a Read Record command was represented which extracted specific data from the card. This data encompassed the Application Effective Date, the Application

Expiration Date, Primary Account Number, PAN Sequence Number, Application Usage Control, Cardholder Verification Method (CVM) List, Issuer Action Codes, and Issuer Country Code. The complete details of the response 12 can be found in **Appendix 1.12**.

Request 13 [31]**: TPDU: 00 b2 02 1c 00 6c 4b**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 2) from the current file (SFI 3). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 75 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback. The complete details of the request 13 can be found in **Appendix 1.13**.

Response 13 [31] **: TPDU: 00 b2 02 1c 4b 70 49 8c 15 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 8d 19 8a 02 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 91 08 9f 08 02 00 a0 5f 30 02 02 01 9f 42 02 08 26 9f 44 01 02 9f 4a 01 82 90 00**

As depicted by the author, a Read Record command was represented that retrieved specific details from the card. This encompassed card risk management data, application version number, service code, application currency code, application currency exponent, and the static data authentication tag list. The complete details of the response 13 can be found in **Appendix 1.13**.

Request 14 [31]**: TPDU: 00 88 00 00 04 ca 2a 88 88 61 87**

As indicated by the author, an internal authenticate command message was presented, containing an unpredictable number. The complete details of the request 14 can be found in **Appendix 1.14**.

Response14 [31]**: TPDU: 00 c0 00 00 87 77 81 84 9f 4b 81 80 86 28 31 3f 3b d9 b9 90 48 0e 31 1c 96 a6 6f 77 d9 9a 79 99 19 bf e9 ae 85 bf fa 1a 77 95 cd 52 7d 4f 5f c3 4d 2d 98 17 84 40 43 d8 c0 75 af b5 f2 81 83 f8 ec 64 ff 19 13 41 e7 df 65 90 a0 5f 0d f9 91 98 99 34 b3 83 5c 22 b4 6f 4b c5 6d 54 09 e5 05 d0 b1 f4 7c 14 f0 a8 e0 bd a8 3c 73 b5 fe 22 cf 8b 68 80 df e8 43 fa 30 77 67 d6 1d 2f 20 8c f7 8d 91 59 1e ed e0 d4 6f 42 e1 9d 8d 4d 90 00**

Based on the author's observations, a get response command was utilized to retrieve the signed dynamic application data. The complete details of the response 14 can be found in **Appendix 1.14**.

Request 15 [31]**: TPDU: 80 ca 9f 17 00 6c 04**

According to the author's analysis, a get data command was employed to retrieve the PIN try counter. The complete details of the request 15 can be found in **Appendix 1.15**.

Response 15 [31]**: TPDU: 80 ca 9f 17 04 9f 17 01 03 90 00**

The author noted that a get data command was issued, subsequently followed by data indicating the PIN try counter. The complete details of the response 15 can be found in **Appendix 1.15**.

Request 16 [31]**: TPDU: 00 20 00 80 08 24 23 07 ff ff ff ff ff 90 00**

As identified by the author, the command depicted signifies a verify command, typically employed to authenticate a PIN or password. The complete details of the request 16 can be found in **Appendix 1.16**.

Request 17 [31]**: TPDU: 00 c0 00 00 20 77 1e 9f 27 01 80 9f 36 02 00 04 9f 26 08 03 e7 e6 aa f0 9d ca 1a 9f 10 07 06 05 12 03 a4 00 02 90 00**

In the observation of the author, the command illustrated was recognized as a get response command tasked with retrieving vital data components such as Cryptogram Information Data (CID), Application Transaction Counter (ATC), Application Cryptogram (AC), and Issuer Application Data (IAD). The complete details of the request 17 can be found in **Appendix 1.17**.

Request 18 [31]**: TPDU: 80 ae 40 00 27 30 30 00 00 00 00 01 00 00 00 00 00 00 00 08 26 00 80 00 80 00 08 26 23 08 03 00 ca 2a 88 88 7b 21 5d b0 00 82 00 00 61 20**

As observed by the author, the presented command was identified as a generate ac command. The complete details of the request 18 can be found in **Appendix 1.18**.

Response 18 [31]**: TPDU: 00 c0 00 00 20 77 1e 9f 27 01 40 9f 36 02 00 04 9f 26 08 fa 09 49 2d c3 cb eb b2 9f 10 07 06 05 12 03 64 00 02 90 00**

As identified by the author, the command in question was a get response command intended to retrieve Cryptogram Information Data (CID), Application Transaction Counter (ATC), Application Cryptogram (AC), and Issuer Application Data (IAD). The complete details of the response 18 can be found in **Appendix 1.18**.

5.2.2.2 MITM ATTACK ON CARD 2 (BARCLAYS VISA CARD)

ATR: 3b 6e 00 00 80 31 80 66 b0 84 0c 01 6e 01 83 00 90 00

The ATR is a message that was output by a contact smart card upon initial powering, as observed by the author, in response to an RST signal sent from the card machine. Information about the card's properties and communication parameters was noted by the author to be contained within it.

Request 1 [30]**: TPDU: 00 a4 04 00 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 30 31 61 1c**

The select command for an EMV application directory was identified by the author. This command, sent from the terminal to the card, requests the card to select the EMV Payment System Directory (PSE), which, as observed by the author, is 1PAY.SYS.DDF01.

Response 1 [30]**: TPDU 00 c0 00 00 1c 6f 1a 84 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 30 31 a5 08 88 01 01 5f 2d 02 65 6e 90 00**

The get response command, which asked for 28 bytes of data related to a previously sent command (request 1), was identified by the author. The response, as was observed by the author, provided details of the 1PAY.SYS.DDF01 EMV application directory, including its priority and language preference (English).

Request 2 [30]**: TPDU: 00 b2 01 0c 00 6c 1a**

The author identified the read record command, which had requested the card to read a specific record (record 1) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 26 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 2 [30]**: TPDU: 00 b2 01 0c 1a 70 18 61 16 4f 07 a0 00 00 00 29 10 10 50 08 4c 49 4e 4b 20 41 54 4d 87 01 01 90 00**

The data returned by the card in response to Request 2 was analyzed by the author and was found to include various information. An application identifier (4f 07 a0 00 00 00 29 10 10) for a specific card application and the application label LINK ATM were among the details noted by the author.

Request 3 [30]**: TPDU: 00 b2 02 0c 00 6c 29**

The author identified the read record command, which had requested the card to read a specific record (record 2) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 41 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 3 [30]**: TPDU: 00 b2 02 0c 16 70 14 61 12 4f 07 a0 00 00 00 29 10 10 50 04 4c 49 4e 4b 87 01 01 90 00**

The data that was returned by the card in response to the command (Request 3) was analyzed by the author and was found to include various details. An application identifier (4f 07 a0 00 00 00 29 10 10) for a specific card application and 50 04 4c 49 4e 4b, which the author identified as standing for the application label (Visa Debit), were among the details noted. The end of the TPDU (90 00), as observed by the author, indicated a successful operation or no error.

Request 4 [30]**:TPDU: 00 b2 03 0c 00 6c 1a**

The author identified the read record command, which had requested the card to read a specific record (record 3) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 26 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 4 [30]**: TPDU: 00 b2 03 0c 1a 70 18 61 16 4f 07 a0 00 00 00 03 80 02 50 08 42 41 52 43 4c 41 59 53 87 01 00 90 00**

The data that was returned by the card in response to the command (Request 4) was analyzed by the author and was found to include information about an application called BARCLAYS. Which is in this case the bank name The end of the TPDU (90 00), as observed by the author, indicated a successful operation or no error.

Request 5 [30]**:TPDU: 00 b2 04 0c 00 6a 83**

The author identified the read record command, which had requested the card to read a specific record (record 4) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested record did not exist.

Request 6 [30]**: TPDU: 00 a4 04 00 07 a0 00 00 00 03 10 10 61 2f**

The select command, which was asking the card to select the Visa Debit application, was identified by the author. It was noted by the author that the card responded by indicating that there were 47 more bytes of data available, which could be obtained by sending a GET RESPONSE command.

Response 6 [30]**: TPDU: 00 c0 00 00 2f 6f 2d 84 07 a0 00 00 00 03 10 10 a5 22 50 0a 56 69 73 61 20 44 65 62 69 74 87 01 02 5f 2d 02 65 6e bf 0c 0b 9f 0a 08 00 01 05 01 00 00 00 00 90 00**

The get response command, which represented the response of a previously issued select command (Request 6), was identified by the author. The author noted that it provided details of the VISA DEBIT application on the card.

Request 7 [30]**: TPDU: 80 a8 00 00 02 83 00 61 14**

This was identified by the author as a get processing options command that had been issued without any specific processing options requested. The author noted that the response status indicated that there were still 20 additional response bytes available for this command.

Response 7 [30]**: TPDU: 00 c0 00 00 14 80 12 38 00 08 01 01 00 08 03 04 00 10 01 02 00 18 01 02 01 90 00**

The get response command, which represented the response of a previously issued get processing options command (Request 7), was identified by the author. It was observed by the author that this command retrieved a response message with details about the card's capabilities and records to be read.

Request 8 [30]**: TPDU: 00 b2 01 0c 00 6c 38**

The author identified the read record command, which had requested the card to read a specific record (record 1) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 56 bytes. This discrepancy likely arose because the terminal might not have known the exact

length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 8 [30]**: 00 b2 01 0c 38 70 36 57 13 46 58 65 47 45 17 60 17 d2 80 82 21 51 40 00 00 00 00 1f 5f 20 03 20 2f 20 9f 1f 18 30 30 30 30 30 30 30 30 30 30 30 30 30 30 35 31 34 30 30 30 30 30 30 90 00**

The data that was read from the card, as noted by the author, contained track data, the cardholder's name, and other discretionary data. It was observed by the author that the READ RECORD command had been successful, and the card had provided the requested data.

Request 9 [30]**: TPDU: 00 b2 03 0c 00 6c fe**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 03) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 254 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 9 [30]**: TPDU: 00 b2 03 0c fe 70 81 fb 90 81 f8 53 38 7e 4d a7 cc 61 53 15 13 b3 39 55 64 ae 71 1a 9f d2 9c 4b c2 03 a1 6d da 8c 64 1a 93 a8 af fa 78 a8 9c 82 cf 0f 3c b7 20 c3 4e 88 d5 26 1d 60 c2 51 c1 4e d6 e7 77 3b bc 63 53 40 72 c3 c4 0b cc 76 83 59 80 1d 6a 32 6c 50 02 ef 52 fc 07 12 65 81 cf 77 5c 17 9e 59 fe e4 38 c7 6b 1a 20 f1 73 d9 36 03 27 3c ac 06 e6 be 08 20 d8 06 2a 9d 37 a9 be e5 df 48 ab 1e f6 2f 7e 09 26 3e 27 e4 b4 d9 06 27 ff 38 61 32 b1 78 a8 5c 9c 24 a8 ad 68 dc 9b 7f ed e6 df 29 e8 df 51 73 69 91 0c 5a da 5c ff 3b 10 fe 79 f8 ad 91 59 be b3 dc ac 15 c8 27 5f 75 9e 7c 09 5b 32 d2 21 45 ef 29 db 0f 8e b1 7f 83 e7 ae 2c 30 59 de 96 1b 44 18 0c 63 b5 1b 7d 97 7d f5 32 c9 b3 96 af 68 8a d6 28 78 28 19 98 77 77 6f f4 3d 74 be 82 dc df ba 0a f3 66 b4 97 4c 52 0d 20 90 00**

The command, identified by the author, was used to read the record. It was specifically aimed at retrieving the Issuer Public Key Certificate, as noted by the author.

Request 10 [30]**: TPDU: 00 b2 04 0c 00 6c b7**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 4) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length

of 183 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 10 [30]**: TPDU: 00 b2 04 0c b7 70 81 b4 9f 46 81 b0 50 e9 28 6b aa 6e 4d d7 2c ff ac 62 61 5d f7 9b 1e ea d8 6a 50 cc 81 6d ed da bc 56 18 8d d0 d0 22 a0 74 41 35 f2 5a ae 44 fb a0 1f 9f ac 90 49 32 ca 06 2f 58 4d 3c 0b 44 23 8e fd 35 0a 07 f4 ec 55 f1 cc 63 12 78 d0 12 27 97 37 61 e7 dd 0c 39 14 ba 49 9f 9f 56 a0 a4 67 73 bb 1d f4 2c 93 c0 d2 71 0c 33 15 f6 2e 1d 59 c2 49 72 e3 b0 41 03 68 c8 08 35 b7 55 6e dc 1a 32 36 c0 3b 26 cc 92 9c 3a 06 b9 b7 72 93 3e 6d c2 4e 3b ac 0f 0b 8e ab 9d 17 8e de 43 2b 44 b8 2d bc f4 86 b6 da 2c 11 54 31 bb b8 e1 b5 d5 22 a3 c9 c9 b6 65 7a 90 00**

The command, identified by the author, was used to read the record. It was specifically aimed at retrieving the Public Key Certificate from the Integrated Circuit Card (ICC), as noted by the author.

Request 11 [30]**: TPDU: 00 b2 01 14 00 6c 19**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 1) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 25 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 11 [30]**: TPDU: 00 b2 01 14 19 70 17 9f 47 01 03 9f 48 0a 4f 88 af 39 25 19 4f 91 b2 0f 9f 49 03 9f 37 04 90 00**

This was identified as a response message by the author, which initiated the card authentication process, as noted in the author's observations.

Request 12 [30]**: TPDU: 00 b2 02 14 00 6c 09**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 2) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 9 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card

ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 12 [30]**: TPDU: 00 b2 02 14 09 70 07 9f 32 01 03 8f 01 09 90 00**

The command, as identified by the author, represented a Read Record command aimed at retrieving data from the card, such as the Certification Authority Public Key Index, which was 9, and the Issuer Public Key Exponent value, which was 3.

Request 13 [30]**: TPDU: 00 b2 01 1c 00 6c 5d**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 1) from the current file (SFI 3). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 9 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 13 [30]**: TPDU: 00 b2 01 1c 5d 70 5b 5f 25 03 23 08 01 5f 24 03 28 08 31 5a 08 46 58 65 47 45 17 60 17 5f 34 01 00 9f 07 02 ff 80 8e 10 00 00 00 00 00 00 00 00 41 03 1e 03 02 03 1f 03 9f 0d 05 b8 60 6c 08 00 9f 0e 05 00 10 80 00 00 9f 0f 05 b8 68 7c 98 00 5f 28 02 08 26 9f 42 02 08 26 9f 44 01 02 9f 4a 01 82 90 00**

The command was identified by the author as representing a Read Record command aimed at retrieving various data from the card, including the Application Effective Date, the Application Expiration Date, Primary Account Number, PAN Sequence Number, Application Usage Control, Cardholder Verification Method (CVM) List, Issuer Action Codes, and Issuer Country Code.

Request 14 [30]**: TPDU: 00 b2 02 1c 00 6c 41**

The author identified the READ RECORD command, which had requested the card to read a specific record (record 2) from the current file (SFI 3). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 65 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 14 [30]**: TPDU: 00 b2 02 1c 41 70 3f 8c 15 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 8d 1c 91 08 8a 02 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 9f 34 03 5f 30 02 02 21 9f 08 02 00 a0 90 00**

As identified by the author, it represented a Read Record command that retrieved data from the card such as card risk management data, application version number, service code, application currency code, application currency exponent, and static data authentication tag list.

Request 15 [30]**: TPDU: 00 88 00 00 04 b7 c6 5c b8 61 93**

This was identified by the author as an internal authenticate command message containing an unpredictable number.

Response 15 [30]**: TPDU: 00 c0 00 00 93 80 81 90 89 e6 da f7 89 b9 d4 2c 9a 3f 54 28 94 13 a8 f0 66 e7 6f cf 76 2d 45 d4 c6 bb 9a 11 25 a3 e7 19 af e7 50 cb 39 81 06 ce d4 88 ac 30 01 13 50 15 09 18 d4 a4 66 01 19 44 48 6e 7d a7 c2 db bf c2 e4 21 b2 78 d6 44 30 af ac 54 6b 61 6c 1f 00 83 d9 fc b6 96 4a 86 ee ec fc ce 42 07 ff 13 71 b8 68 8b c3 8c 76 b4 ae aa 23 db f9 20 de 66 28 9f 3e 04 09 65 b5 86 37 23 24 45 cf 17 5d 5d 9e 63 77 d7 57 31 a8 7a 44 df ba c2 e0 51 b0 4c 64 c1 90 00**

This was identified by the author as a get response command aimed at retrieving the signed dynamic application data.

Request 16 [30]**: TPDU: 80 ca 9f 17 00 6c 04**

According to the author's analysis, a get data command was employed to retrieve the PIN try counter.

Response 16 [30]**: TPDU: 80 ca 9f 17 04 9f 17 01 03 90 00**

The author noted that a get data command was issued, subsequently followed by data indicating the PIN try counter.

Request 17 [30]**: TPDU: 00 20 00 80 08 24 23 07 ff ff ff ff ff 90 00**

As identified by the author, the command depicted signifies a verify command, typically employed to authenticate a PIN or password.

Request 18 [30]**: TPDU: 80 ae 80 00 1d 00 00 00 00 01 00 00 00 00 00 00 00 08 26 00 80 00 80 00 08 26 23 09 04 00 b7 c6 5c b8 61 14**

As observed by the author, the presented command was identified as a generate ac command (first cryptogram request).

Response 18 [30]**: TPDU: 00 c0 00 00 14 80 12 80 00 46 cc 6c 03 71 b9 85 b8 ed 06 0e 12 03 a4 20 02 90 00**

As identified by the author, the command in question was a get response command intended to retrieve Cryptogram Information Data (CID), Application Transaction Counter (ATC), Application Cryptogram (AC), and Issuer Application Data (IAD).

Request 19 [30]**: 80 ae 40 00 2a 54 f2 09 c8 00 84 00 00 30 30 00 00 00 00 01 00 00 00 00 00 00 00 08 26 00 80 00 80 00 08 26 23 09 04 00 b7 c6 5c b8 41 03 02 61 14**

As observed by the author, the presented command was identified as a generate ac command (second cryptogram request).

Response 19 [30]**: TPDU: 00 c0 00 00 14 80 12 80 00 46 cc 6c 03 71 b9 85 b8 ed 06 0e 12 03 a4 20 02 90 00**

As identified by the author, the command in question was a get response command intended to retrieve Cryptogram Information Data (CID), Application Transaction Counter (ATC), Application Cryptogram (AC), and Issuer Application Data (IAD).

5.2.2.3 MITM ATTACK ON CARD 3 (METRO BANK MASTERCARD)

ATR: ATR: 3b 6e 00 00 80 31 80 66 b0 84 0c 01 6e 01 83 00 90 00

The ATR is a message that was output by a contact smart card upon initial powering, as observed by the author, in response to an RST signal sent from the card machine. Information about the card's properties and communication parameters was noted by the author to be contained within it.

Request 1 [30]**: TPDU: 00 a4 04 00 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 30 31 61 20**

The select command for an EMV application directory was identified by the author. This command, sent from the terminal to the card, requests the card to select the EMV Payment System Directory (PSE), which, as observed by the author, is 1PAY.SYS.DDF01.

Response 1 [30]**: TPDU 00 c0 00 00 20 6f 1e 84 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 30 31 a5 0c 88 01 01 5f 2d 02 65 6e 9f 11 01 01 90 00**

The get response command, which asked for 32 bytes of data related to a previously sent command (request 1), was identified by the author. The response, as was observed by the author, provided details of the 1PAY.SYS.DDF01 EMV application directory, including its priority and language preference (English).

Request 2 [30]**: TPDU: 00 b2 01 0c 00 6c 38**

The author identified the read record command, which had requested the card to read a specific record (record 1) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 56 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 2 [30]**: TPDU: 00 b2 01 0c 38 70 36 61 34 4f 07 a0 00 00 00 04 10 10 50 0a 4d 61 73 74 65 72 63 61 72 64 9f 12 10 44 65 62 69 74 20 4d 61 73 74 65 72 63 61 72 64 87 01 02 73 07 9f 0a 04 00 01 01 01 90 00**

The data returned by the card in response to Request 2 was analyzed by the author and was found to include various information. An application identifier (4f 07 a0 00 00 00 04 10 10) for a specific card application and the application label Mastercard (Debit Mastercard) were among the details noted by the author.

Request 3 [30]**: TPDU: 00 b2 02 0c 00 6c 16**

The author identified the read record command, which had requested the card to read a specific record (record 2) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 22 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 3 [30]**: TPDU: 00 b2 02 0c 16 70 14 61 12 4f 07 a0 00 00 00 29 10 10 50 04 4c 49 4e 4b 87 01 01 90 00**

The data that was returned by the card in response to the command (Request 3) was analyzed by the author and was found to include various details. An application identifier (4f 07 a0 00 00 00 00 29 10 10) for a specific card application and 50 04 4c 49 4e 4b, which the author identified as standing for the application label (LINK), were among the details noted. The end of the TPDU (90 00), as observed by the author, indicated a successful operation or no error.

Request 4 [30]**:TPDU: 00 b2 03 0c 00 6a 83**

The author identified the read record command, which had requested the card to read a specific record (record 3) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested record did not exist OR not allowed to access.

Request 5 [30]**:TPDU: 00 a4 04 00 07 a0 00 00 00 04 10 10 61 51**

The select command, which was asking the card to select the Debit Mastercard application, was identified by the author. It was noted by the author that the card responded by indicating that there were 81 more bytes of data available, which could be obtained by sending a get response command.

Response 5 [30]**: TPDU: 00 c0 00 00 51 6f 4f 84 07 a0 00 00 00 04 10 10 a5 44 50 0a 4d 61 73 74 65 72 63 61 72 64 87 01 02 5f 2d 02 65 6e 9f 11 01 01 9f 12 10 44 65 62 69 74 20 4d 61 73 74 65 72 63 61 72 64 bf 0c 16 9f 4d 02 0b 0a 9f 6e 07 08 26 00 00 30 30 00 9f 0a 04 00 01 01 01 90 00**

The get response command, which represented the response of a previously issued select command (Request 5), was identified by the author. The author noted that it provided details of the Debit Mastercard application on the card.

Request 6 [30]**: TPDU: 80 a8 00 00 02 83 00 61 10**

This was identified by the author as a get processing options command that had been issued without any specific processing options requested. The author noted that the response status indicated that there were still 16 additional response bytes available for this command.

Response 6 [30]**: TPDU: 00 c0 00 00 10 77 0e 82 02 39 00 94 08 08 02 03 01 10 01 04 00 90 00**

The get response command, which represented the response of a previously issued get processing options command (Request 6), was identified by the author. It was observed by the author that this command retrieved a response message with details about the card's capabilities and records to be read.

Request 7 [30]**: TPDU: 00 b2 02 0c 00 6c 98**

The author identified the read record command, which had requested the card to read a specific record (record 2) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 152 bytes. This discrepancy likely arose because the terminal might not have known the exact

length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 7 [30]**: 00 b2 02 0c 98 70 81 95 9f 42 02 08 26 5f 25 03 23 07 01 5f 24 03 26 09 30 5a 08 55 73 61 50 53 18 62 93 5f 34 01 02 9f 07 02 ff c0 8c 2a 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 9f 35 01 9f 45 02 9f 4c 08 9f 34 03 9f 21 03 9f 7c 14 9f 1d 08 8d 0c 91 0a 8a 02 95 05 9f 37 04 9f 4c 08 8e 14 00 00 00 00 00 00 00 00 42 01 44 03 41 03 02 03 1e 03 1f 03 9f 0d 05 bc 50 bc 88 00 9f 0e 05 00 00 00 00 00 9f 0f 05 bc 70 bc 98 00 5f 28 02 08 26 9f 4a 01 82 90 00**

The command was identified by the author as representing a Read Record command aimed at retrieving various data from the card, including the Application Effective Date, the Application Expiration Date, Primary Account Number, PAN Sequence Number, Application Usage Control, Cardholder Verification Method (CVM) List, Issuer Action Codes, and Issuer Country Code.

Request 8 [30]**: TPDU: 00 b2 03 0c 00 6c 44**

The author identified the read record command, which had requested the card to read a specific record (record 3) from the current file (SFI 1). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 68 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 8 [30]**: TPDU: 00 b2 03 0c 44 70 42 57 10 55 73 61 50 53 18 62 93 d2 60 92 01 90 20 07 99 5f 20 0a 4c 49 44 48 45 45 4d 2f 4d 20 5f 30 02 02 01 9f 49 03 9f 37 04 9f 08 02 00 02 9f 1f 10 39 30 32 30 30 30 30 37 39 39 30 30 30 30 30 30 90 00**

The data that was read from the card, as noted by the author, contained track data, the cardholder's name, and other discretionary data. It was observed by the author that the read record command had been successful, and the card had provided the requested data.

Request 9 [30]**: TPDU: 00 b2 01 14 00 6c fe**

The author identified the read record command, which had requested the card to read a specific record (record 1) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 254

bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 9 [30]**: TPDU: 00 b2 01 14 fe 70 81 fb 90 81 f8 c6 82 5c 30 83 45 64 e5 93 73 1c ad 6b 21 5b aa ea 43 ca e2 d5 62 c5 45 ed 4c 73 16 bd ff b4 1f 42 24 69 63 2e 18 4a 9c ca 5d a3 29 da 57 8d 7c db 87 8c 2a ab fd a7 f1 40 dc 9e a5 b9 df bc 20 ce 1e 59 00 8a d1 18 9c 87 54 4e 90 1a 19 95 bd 18 f9 5a 33 59 8e 5b 96 fd 4e 59 c4 1a 75 3a 91 ac 49 54 cb 12 94 e2 5e f1 1a 4b e6 af cc a1 8a 65 4d 19 ef 78 18 f1 60 a4 72 82 2c e1 92 24 b1 d8 ff 87 ff db fd c7 89 79 1c a7 86 fe 16 be df 1b 5d 0d ad 1b c5 4e 18 16 1b 93 ec 50 2d 47 b0 dd 98 76 f3 e8 6c 01 08 87 18 11 27 78 e7 14 8d 1d fc 7c 3a c1 32 56 d3 16 69 0c 86 92 84 c1 84 4c 60 83 ad fe 96 c2 ee 3f 35 73 6f 87 52 0d 28 51 13 13 2f 57 32 4e 15 4e 51 bd 25 9c b8 cd b6 71 af 28 e9 c9 66 e8 c7 fc 4b 3b 94 43 7b 39 11 da 94 04 6a 59 7e f5 b9 90 00**

The command, identified by the author, was used to read the record. It was specifically aimed at retrieving the Issuer Public Key Certificate, as noted by the author.

Request 10 [30]**: TPDU: 00 b2 02 14 00 6c 09**

The author identified the read record command, which had requested the card to read a specific record (record 2) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 9 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 10 [30]**: TPDU: 00 b2 02 14 09 70 07 8f 01 06 9f 32 01 03 90 00**

The command, as identified by the author, represented a read record command aimed at retrieving data from the card, such as the Certification Authority Public Key Index, which was 6, and the Issuer Public Key Exponent value, which was 3.

Request 11 [30]**: TPDU: 00 b2 03 14 00 6c 33**

The author identified the read record command, which had requested the card to read a specific record (record 3) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 51 bytes. This discrepancy likely arose because the terminal might not have known the exact

length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 11 [30]**: TPDU: 00 b2 03 14 33 70 31 9f 47 01 03 9f 48 2a c3 2c f0 d5 ce 6c 87 58 95 18 27 9d 04 53 be 19 e7 45 10 68 72 7a 95 2a 12 89 af 2f 39 67 1b 2b ed 90 d1 6c 98 ef 43 6a db 9d 90 00**

The command, as identified by the author, represented a read record command aimed at retrieving data from the card, such as the Integrated Circuit Card (ICC) Public Key Exponent, and the Integrated Circuit Card (ICC) Public Key Remainder.

Request 12 [30]**: TPDU: 00 b2 04 14 00 6c b7**

The author identified the read record command, which had requested the card to read a specific record (record 4) from the current file (SFI 2). The card's response, as observed by the author, indicated that the requested data length was incorrect and provided the correct length of 183 bytes. This discrepancy likely arose because the terminal might not have known the exact length of a record or data in advance. As noted by the author, such feedback from the card ensured more seamless interactions, permitting the terminal to modify its commands based on this real-time feedback.

Response 12 [30]**: TPDU: 00 b2 04 14 b7 70 81 b4 9f 46 81 b0 b2 6f e8 f0 66 46 ef e7 28 1e a5 10 cb 98 2d af 14 7e 88 35 5e b4 32 cd d4 92 ea f0 43 da 19 22 85 1e 4d ea 7d 1d 08 97 0e e4 e3 e8 c7 50 c9 fd 52 43 48 ce ca 6d 64 98 19 32 82 e2 ae 8a 2e 3b ee 7a 27 d7 fb b5 b9 c0 46 ff c2 a3 7a 79 dd a8 76 87 ad 22 5c 1e 65 16 fd 71 87 1c 55 1a 2f ac ef bc bc 8c 49 17 32 d7 e3 48 02 3f 49 bc a6 7f 96 f0 f7 26 58 24 f7 bf 32 a9 44 f4 80 59 71 2a ad 22 50 31 ee 39 ff 51 1b eb 2a a1 e8 f5 2b 91 b3 83 8c 72 c8 57 6a 45 10 18 ab 85 e0 bb 48 66 2b 16 09 60 c1 76 0f 07 a2 90 63 56 5d de a2 7f 90 00**

The command, identified by the author, was used to read the record. It was specifically aimed at retrieving the Public Key Certificate from the Integrated Circuit Card (ICC), as noted by the author.

Request 13 [30]**: TPDU: 80 ca 9f 17 00 6c 04**

According to the author's analysis, a get data command was employed to retrieve the PIN try counter.

Response 13 [30]**: TPDU: 80 ca 9f 17 04 9f 17 01 03 90 00**

The author noted that a get data command was issued, subsequently followed by data indicating the PIN try counter.

Request 14 [30]**: TPDU: 00 84 00 00 00 6c 08**

According to the author's analysis, a get challenge was employed.

Response 14 [30]**: TPDU: 00 84 00 00 08 a6 4c 8b a6 4f a7 76 82 90 00**

The author noted that a get data command was issued, subsequently followed by data indicating unpredictable number generated by the ICC.

Request 15 [30]**: 00 20 00 88 b0 6e a9 8f c7 27 8d 6e 53 50 f9 ff 87 4a 40 b0 61 23 11 3f be 27 75 16 50 f4 7a 14 7d fc c3 5d 45 0c 88 a7 05 9a e9 6a 5e 9a 55 a4 72 4d e2 39 6a 11 4b 4b 65 72 99 3d b3 79 6c 2d 0f ed 71 98 8d 39 d2 a3 e4 22 85 84 22 53 9a 67 6c b0 fd 1c 0b b2 42 ab 8f 56 ad ea c9 fa 02 d5 7b f5 8b 21 cc a5 19 f8 68 99 86 87 d6 86 f6 0d f8 bb 07 43 69 0e 7e 64 74 00 60 27 04 ac 16 54 38 a6 de 1f 65 85 8f 27 cd 13 81 6e 3a af 11 22 a4 dd 68 b1 44 3b 3c 27 e8 7a f1 a4 fc 71 62 08 e3 87 34 24 b1 26 4f 22 61 fb 1f 88 b4 fb c2 15 26 e5 d3 9a ae 90 00**

As identified by the author, the command depicted signifies a verify command, typically employed to authenticate a PIN or password and in this case its encrypted PIN.

Request 16 [30]**: TPDU: 80 ae 90 00 4a 00 00 00 00 01 00 00 00 00 00 00 00 08 26 00 00 00 80 00 08 26 23 09 04 00 56 31 62 e4 22 00 00 00 00 00 00 00 00 00 00 44 03 02 14 27 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 61 d5**

As observed by the author, the presented command was identified as a generate ac command (first cryptogram request).

Response 16 [30]**: TPDU: 00 c0 00 00 d5 77 81 d2 9f 27 01 80 9f 36 02 00 03 9f 4b 81 b0 9d ab 36 d8 31 78 f1 4c 2c 0b 63 9d 9d bf fd 17 fb a1 38 45 be db 53 4a 1e a4 fe 46 43 05 98 bd 4a 24 13 22 12 02 01 cc e9 9d 74 7b 8d 81 3d e1 57 20 9b 37 ad a8 05 08 5e f5 b2 da f3 a3 c8 be 9b 52 fe 32 9c 04 81 bd 69 01 86 6a 07 cd ef 86 dd 2a b5 21 3b 96 a3 8f 66 a5 f2 eb f2 cb 5e 84 da a2 60 59 31 95 dd 5a c5 61 6a 6c c3 87 7c 7f f1 6b 9c 88 06 09 d0 ed 7d 86 f8 a9 e3 7b e7 67 d0 e1 c4 de 9d 0f 3b 7b 10 d0 08 58 93 fc 4b 29 f9 13 44 a9 83 4a 6e d2 5e e4 0b b8 8b 92 e1 ba 62 73 2e 62 26 8a 9c 50 e1 b8 1b 70 7a 40 e3 3f 9f 10 12 03 10 a7 40 03 02 00 00 00 00 00 00 00 00 00 00 ff 90 00**

As identified by the author, the command in question was a get response command intended to retrieve Cryptogram Information Data (CID), Application Transaction Counter (ATC), Application Cryptogram (AC), and Issuer Application Data (IAD).

Request 17 [30]**: TPDU: 80 ae 50 00 1d e1 8e 9b 81 4d 7c 2c e4 00 10 30 30 00 00 00 80 00 56 31 62 e4 1e 1f c6 38 a5 da 01 94 61 d5**

As observed by the author, the presented command was identified as a generate ac command (second cryptogram request).

Response 17 [30]**: TPDU: 00 c0 00 00 d5 77 81 d2 9f 27 01 40 9f 36 02 00 03 9f 4b 81 b0 76 86 9b 59 b0 d9 a7 13 a9 e9 e6 eb 03 64 d6 a1 ae 44 b9 4a 5d 47 67 4e eb de 90 d6 a3 a6 d4 82 a0 38 3e 1c 74 70 de fc 93 74 45 a0 b4 39 4f e2 90 d5 4c ba 9d 13 12 bc bc dd 45 bd 43 69 32 35 26 da 2a 40 fd cd 6c 86 10 7d c1 fb a8 88 30 e3 43 27 68 5f 84 81 55 df f7 52 d6 b3 51 79 15 22 54 a6 65 a5 03 b9 2a 05 20 dd b2 96 db 3d 91 3f 9a e1 8f 60 03 ce 14 0a 51 4a 89 83 5c ae 98 62 99 2c 06 60 30 2d f3 f5 96 9e f3 23 47 db 13 20 73 8b 29 8e c2 a3 0e 32 dc db e5 39 02 16 cf a2 f1 23 89 2d d7 64 40 4a de 6a 7a 75 15 26 90 7e 9f 10 12 03 10 67 70 03 02 00 00 1e 1f 00 00 00 00 00 00 00 ff 90 00**

As identified by the author, the command in question was a get response command intended to retrieve Cryptogram Information Data (CID), Application Transaction Counter (ATC), Application Cryptogram (AC), and Issuer Application Data (IAD).

5.2.3 SUMMARY

During the MITM attack, the author managed to intercept 18 requests and 15 responses for each card. By analyzing this data, the author obtained crucial details about the card and transaction. These details were correct at the time of the experiment. For the safety of the author, the cards in use will be deactivated and a replacement will be obtained from the bank. Intercepted cards details will be presented in the table 2.

MITM Attack	Status	Outcome	Mitigation
Card 1 (Visa) Halifax bank	Successful	• Track 2 Equivalent Data: Yes • Track 1 Discretionary Data: Yes • Cardholder Name: Gained • Application effective date: Yes • Application expiration: Yes • Card number: Yes • PIN code: Yes, plaintext PIN (unencrypted)	• Cardholder should receive real time notification of any transaction even if its low-risk transaction. • Card issuing bank should be signing the CVM list rule and give PIN verification the highest priority. • Card issuing bank should choose enciphered pin verification even if it might be slower due to encryption and decryption.
Card 2 (Visa) Barclays bank	Successful	• Track 2 Equivalent Data: Yes • Track 1 Discretionary Data: Yes • Cardholder Name: Yes • Application effective date: Yes • Application expiration: Yes • Card number: Yes • PIN code: Yes, Plaintext PIN (unencrypted)	• Cardholder should receive real time notification of any transaction even if its low-risk transaction. • Card issuing bank should be signing the CVM list rule and give PIN verification the highest priority. • Card issuing bank should choose enciphered pin verification even if it might be slower due to encryption and decryption.
Card 3 (Mastercard) Metro bank	Successful	• Track 2 Equivalent Data: Yes • Track 1 Discretionary Data: Yes • Cardholder Name: Yes • Application effective date: Yes • Application expiration: Yes • Card number: Yes • PIN code: No, Enciphered PIN (encrypted)	• Cardholder should receive real time notification of any transaction even if its low-risk transaction. • Card issuing bank should be signing the CVM list rule and give PIN verification the highest priority.

Table 2: MITM Attack on bank cards

Based on observations made by the author during the MITM attack on card 1, card 2, and card 3, as outlined in Table 2, it was identified by the author that all three cards followed the same request and response mechanism. Additionally, it was noted by the author that card 3 (Mastercard) adopted a different approach with regard to the verify command; specifically, it utilized enciphered PIN verification, which was encrypted by the terminal based on the preferences set by the issuing bank (Metro) in the CVM list. Since The central fault in the EMV protocol is that the PIN verification step is never clearly authenticated [32], it's always recommended to employ enciphered PIN verification for offline transaction.

From the observations made by the author, the card details displayed in the table 2 can be cloned onto a physical EMV card using specialized software, such as EMV Reader/Writer v8.6 [33]. This software is available at a price of €2000, but it's also found on the dark web for $250 worth of bitcoin. Due to its high cost, the author couldn't test the software.

Theoretically, software could have the potential to clone a card even without accessing the CVV. This is because the CVV operates outside the boundaries of the EMV protocol (usually the bank is responsible for producing CVV), primarily serving transactions where the card isn't physically present, such as online or phone purchases. Which means that the cloned card can be used only in face-to-face transaction, However, in specific situations where the card is present, the CVV might still be required, especially during identity verifications or AVS checks, ensuring that both the card and its owner correspond to the same registered address.

6 CONCLUSIONS AND RECOMMENDATIONS

6.1 SUMMARY

To put it simply, exploring the security of EMV chip and PIN cards is a complicated task that requires a deep understanding of various EMV protocols, like ISO 7816, ISO 8583, ISO 14443, ISO 16642, ISO 18092, and ISO 20022.

The author's research is divided into two main parts. In the first part, they gathered information about EMV protocols by studying previous research and checking the official EMVCo website. Once they had enough knowledge, they moved on to the second part: conducting experiments. This research is incredibly important as it examines how EMV card transactions are managed. These card transactions are a vital part of global financial services, especially with the increasing use of non-cash payments. If the security of these card payment protocols is compromised, it could have severe and widespread consequences for the world's financial stability.

6.2 CONCLUSION

In this section, the author documented the progress of the experiment and evaluate whether the research goals and objectives have been successfully achieved. In the first chapter, the author detailed the specific aims and objectives of the research, with a primary focus on investigating the security aspects of EMV chip and pin cards. Proceeding to the second chapter, the author conscientiously documented the research phase of the study, marking a notable accomplishment in the research process.

Moving on to chapter three, the author utilized the research onion model to devise a methodology that guided the project's execution. The author acknowledged the model's effectiveness in providing a structured framework for successfully completing the project.

In the fourth chapter, where the focus is on design and implementation, the author meticulously detailed all the necessary steps to set up the experiment. This covered both the software and hardware aspects, and the author also provided the initial design for both pre-experiment and the experiment itself.

Moving into chapter five, it served as a platform for the author to present the discoveries and outcomes obtained from the executed experiment. Within this experimental context, the author conducted three separate tests, with each outcome being thoroughly discussed. An especially significant insight arose from applying a MITM attack: even when an incorrect PIN was entered, a substantial portion of the card's data remained accessible during transactions (most of the details except the PIN code and transaction certificate).

This finding highlights a possible vulnerability: someone could secretly intercept these transactions, acquiring card data to create a duplicate card. This discovery underscores the importance of robust security measures in payment processes.

In Chapter Six, the author provides a comprehensive summary of the entire research, draws conclusions from the experiment, and offers recommendations for future research.

In summary, the author is confident that all the research objectives outlined have been successfully achieved.

6.3 RECOMMENDATIONS FOR FUTURE RESEARCH

The findings of this study, while insightful, open the door for further exploration and examination. Based on the research findings and limitations faced during this study.

The author believes there's potential to enhance the capabilities of the simtrace2 device. This could involve modifying its firmware to enable real-time packet modification. Such an advancement could facilitate more comprehensive testing of the EMV protocol, potentially revealing additional vulnerabilities.

The author is keenly interested in the CVM list, specifically in exploring whether it's possible to modify these lists or adjust their priorities. For instance, can the signature verification be prioritized to come before other methods? Additionally, the author wants to investigate if the PIN attempt counter can be reset.

Another area of interest for the author is delving deeper into the issuer script's functioning. This includes exploring the existence of rules or templates governing issuer scripts, as well as assessing the feasibility of making modifications. It's important to note that these suggestions are solely intended for educational purposes.

It's important to note that these recommendations are based on the present research's findings and the gaps identified. As the field evolves, there may be new areas of interest that future researchers should consider.

7 REFLECTIONS

7.1 INTRODUCTION

As the author reflects upon this project, it's clear how much they've grown both professionally and personally. Each challenge encountered became a lesson, pushing the author to stretch their capabilities and think innovatively. While there were moments of doubt, they were overshadowed by the successes and the knowledge gained. The feedback the author received was a guiding light, directing their efforts and refining their approach. Overall, this experience has not only enriched the author's skills but also instilled a deeper appreciation for the journey of continuous learning.

7.2 LESSON LEARNT

The author has gained several lessons from this research, which may offer insights for future studies. The significance of task management and how it simplifies activities was understood. Prioritizing the tasks, the author began with the research phase, even though they had all the necessary equipment for experimentation. Additionally, the author developed skills in troubleshooting errors and implementing solutions. It was recognized that while a foundational knowledge base is vital, it's crucial to stay updated, particularly in rapidly evolving fields like cybersecurity and computer networking where new vulnerabilities consistently emerge. A final lesson is that with the right allocation of resources and time, any research project, regardless of its scale, can be successfully completed.

7.3 THINGS THAT COULD BE DONE DIFFERENTLY

Regarding this study, certain aspects could have been approached differently:

- Initiating the project earlier than originally planned could have prevented time constraints on specific tasks.
- More time should have been allocated to the experimentation phase, as the author needed additional time to familiarize themselves with the EMV protocol, being new to the subject.
- A deeper understanding of the C language and embedded systems would have enabled the author to reprogram the microcontroller for real-time packet modifications.

- Instead of using a debit card, the author might have considered using a pre-paid MasterCard or Visa card. This could have prevented the situation where one of the test cards was flagged for suspicious activity by the bank, even though the issue was eventually resolved.

7.4 BARRIERS EXPERIENCED

Throughout the project, several barriers were faced. While these challenges presented difficulties, they also provided opportunities for growth, adaptation, and the formulation of new strategies. Below are some of the primary barriers experienced:

- **Lack of Prior Knowledge:** Being new to the subject, it took significant time and resources to understand the details of the EMV protocol. This initial learning curve delayed some aspects of the project.
- **Resource Limitations:** regarding the limitation of the simtrace2 device and limited access to essential tools and the unavailability of specific expertise, resource limitations caused challenges.
- **Time Constraints:** The project could have benefited from an earlier start. Certain tasks took longer than expected, leading to tight schedules.
- **Technical Glitches:** Technical difficulties arose during the project. At times, the entire experiment would stall at the initial stage (entering the main loop). Additionally, hardware malfunctions led to extra bytes of data appearing within the captured TPDU at various points. These issues required prompt attention and backtracking.

Reflecting on these barriers offers valuable insights for evaluating this project and also serves as guidance for future initiatives, helping to navigate similar challenges with greater efficiency.

7.5 POSITIVES

Despite facing various challenges, there were several positive outcomes. Here are some of the significant positives:

- The project provided an opportunity to learn new skills and more understanding in areas previously unfamiliar such as GSM 11.11 and the ISO/IEC 7816 which starts from 1-15 except number 14 which does not exist yet.

- Receiving thorough guidance from the supervisor was essential in producing high-quality work.
- Despite the challenges the author faced, most of the project objectives were achieved.
- The results gained from this project have set the groundwork for continued research and development in the payment system.

7.6 GUIDANCE TO THE FUTURE STUDENTS

After completing this research, the author offers some guidance to future students to assist them in their project research as follows:

- Determining the project's topic early on. The author chooses this topic nearly January this year which is 8 months before the deadline.
- Maintaining motivation and dedication throughout can make a significant difference in the outcome.
- Regularly consult the project supervisor and be open to their guidance. Their experience can provide invaluable insights and steer you in the right direction. The author was consulting his supervisor on weekly basis.
- Allocate tasks based on priority without compromising the quality. Utilize tools or techniques like Gantt charts as illustrated in Appendix 2.
- Stay on top of your timelines. Meeting interim milestones can prevent a last-minute rush and potential oversights. For instance, the author prepared the dissertation draft for review 15 days ahead of the final deadline.
- Comply with all the university regulations and guidelines. As an example, the author submitted the ethical form, as referenced in Appendix 3.

7.7 SUMMARY

The author's research project was a journey of growth and learning. Faced with challenges like time constraints and technical issues, there were also valuable lessons learned about task management and the importance of continuous study in fast-evolving fields. Despite the challenges, there were positive moments, including insights into new areas and guidance from the project supervisor. The author concludes by offering advice to future students on planning, and following university guidelines.

8 REFERENCES

[1] EMVCo, "EMV® Technologies," emvco, 2023. [Online]. Available: https://www.emvco.com/. [Accessed 17 07 2023].

[2] EMVCo, "Worldwide EMV® Deployment Statistics," EMVCo, 2023. [Online]. Available: https://www.emvco.com/about-us/worldwide-emv-deployment-statistics/. [Accessed 12 06 2023].

[3] X. Lan, J. Xu, Z. Zhang, X. Chen and Y. Luo, "A systematic security analysis of EMV protocol," *Computer Standards & Interfaces,* vol. 84, 2023.

[4] U. Finance, "UK Finance," UK Finance org, 2021. [Online]. Available: https://www.ukfinance.org.uk/system/files/Half-year-fraud-update-2021-FINAL.pdf. [Accessed 13 03 2023].

[5] Leigh-Anne Galloway , Tim Yunusov, "First Contact: New vulnerabilities in Contactless Payments," 2019.

[6] Ahmed O. Ibrahim Yaseen Hikmat Ismael, "EMV Electronic Payment System and its Attacks: A Review," *Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM),* vol. 16, no. 1, pp. 23-29, 2022.

[7] Dilpreet Singh, Ron Ruhl and Hamman Samuel, "Attack Tree for Modelling Unauthorized EMV Card Transactions," *4th International Conference on Information Systems Security and Privacy,* 2018.

[8] J. d. R. a. E. Poll, "Formal Analysis of the EMV Protocol Suite," Institute for Computing and Information Science (ICIS) Radboud University Nijmegen, Nijmegen, Netherlands, 2014.

[9] Akeem Olowolayemo#, Nafisat Adewale*, Akram M. Zeki**, Zubair Ahmad, "Examining Users' Understanding of Security Failures in EMV Smart Payment Systems," *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION,* vol. 3, no. 2, pp. 185-191, 2019.

[10] ISO, "ISO/IEC 7816-8:2021(en)," ISO, 2021. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:7816:-8:en. [Accessed 20 05 2023].

[11] EMVCO, "What are EMV® Level 1 and Level 2 Testing," EMVCO, 2023. [Online]. Available: https://www.emvco.com/knowledge-hub/what-are-emv-level-1-and-level-2-testing/. [Accessed 20 05 2023].

[12] ISO, "ISO/IEC 7810:2019," ISO, 2019. [Online]. Available: https://www.iso.org/standard/70483.html. [Accessed 13 08 2023].

[13] UDEMY, "complete-emv-cards-and-payments-training-course," UDEMY, 2022. [Online]. Available: https://www.udemy.com/course/complete-emv-cards-and-payments-training-course/learn/lecture/37636352#overview. [Accessed 27 06 2023].

[14] ISO, "ISO/IEC 7811-1:2018," ISO, 2018. [Online]. Available: https://www.iso.org/standard/73637.html. [Accessed 13 08 2023].

[15] I. Dubinsky, Acquiring Card Payments, NEW YORK: Taylor & Francis Group, 2019.

[16] ISO, "ISO/IEC 7813:2006," ISO, 2006. [Online]. Available: https://www.iso.org/standard/43317.html. [Accessed 13 08 2023].

[17] ISO, "ISO/IEC 4909:2006," ISO, 2006. [Online]. Available: https://www.iso.org/standard/43309.html. [Accessed 13 08 2023].

[18] A. PRÁCE, "SECURITY OF CONTACTLESS SMART CARD PROTOCOLS," BRNO UNIVERSITY OF TECHNOLOGY, BRNO , 2016.

[19] J. Breier, "Cryptography in Payment Systems," Nanyang Technological University, Singapore, 2019.

[20] ISO, "ISO 9564-1:2002," ISO, 2002. [Online]. Available: https://www.iso.org/standard/29374.html. [Accessed 13 08 2023].

[21] J. Luz, F. Schrooyen, G. Ergeerts, R. Beyers, K. Renckens, L. Wante and M. Ceulemans, "A Mobile NFC Payment Terminal for the Event-Wallet on an Android Smartphone," 2023. [Online]. Available: https://www.researchgate.net/figure/APDU-command-and-response-fields_fig3_265061081. [Accessed 25 06 2023].

[22] L. EMVCo, EMV Integrated Circuit Card Specifications for Payment Systems Book 3 Application Specification V 4.3, EMVCo, LLC, 2011.

[23] Andrea Barisani Daniele Bianco Adam Laurie Zac Franken, "Chip & PIN is definitely broken Credit Card skimming and PIN harvesting in an EMV world," Inversepath, 2015.

[24] E. V. A. Dissanayake, "A Systematic Approach for Designing Research Methodology," 2023. [Online]. [Accessed 2023 07 2023].

[25] M. N. S. P. L. A. THORNHILL, in *RESEARCH METHODS FOR BUSINESS STUDENTS EIGHTH EDITION*, Harlow, Pearson Education Limited, 2019.

[26] Viswanath Venkatesh Susan A. Brown Hillol Bala, "RIDGING THE QUALITATIVE–QUANTITATIVE DIVIDE:GUIDELINES FOR CONDUCTING MIXED METHODS RESEARCH IN I NFORMATION SYSTEMS," *MIS Quarterly* , vol. 37, no. 1, pp. 21-54, 2013.

[27] Osmocom, "Osmocom SIMtrace 2," Osmocom, [Online]. Available: https://osmocom.org/projects/simtrace2/wiki. [Accessed 24 03 2023].

[28] Ubuntu, "Ubuntu 22.04.2 LTS (Jammy Jellyfish)," Canonical Ltd, 2018. [Online]. Available: https://releases.ubuntu.com/jammy/. [Accessed 29 07 2023].

[29] W. Foundation, "The world's most popular network protocol analyzer," Wireshark Foundation , 2023. [Online]. Available: https://www.wireshark.org/. [Accessed 29 07 2023].

[30] EMVCo, "Commands," in *Application Independent ICC to Terminal Interface Requirements*, EMVCo, 2000.

[31] E. b. 3, Application Specification book 3, EMVCo.

[32] S. D. R. A. M. B. Steven J. Murdoch, "Chip and PIN is Broken," University of Cambridge, Cambridge, 2010.

[33] E. G. S. INC, "EMV.G.S Software Gallery," Emv Global Solution INC, 2023. [Online]. Available: https://www.emv-global-solution.net/product/emv-reader-writer-v8-6-emv-software/. [Accessed 07 08 2023].

[34] S. Murdoch, "https://emvlab.org/tlvutils/," emvlab.org, 2019. [Online]. Available: https://emvlab.org/tlvutils/. [Accessed 05 08 2023].

[35] ISO, "ISO 8583-1:2003(en) Financial transaction card originated messages," ISO, [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:8583:-1:ed-1:v1:en. [Accessed 06 08 2023].

[36] Danushka Jayasinghe, Raja Naeem Akram, Konstantinos Markantonakis, Konstantinos Rantosy and Keith Mayes, "Enhancing EMV Online PIN Verification," *21st*

International Conference on Financial Cryptography and Data Security (FC '17), pp. 176-195, 2018.

[37] C. S. GmbH, "OpenCard Framework," CardContact Systems GmbH, 2020. [Online]. Available: https://www.openscdp.org/ocf/index.html. [Accessed 04 07 2023].

[38] N. S. Maddi, "EMV Chip and PIN : A Feeble Upgrade," St. Cloud State University, St. Cloud, Minnesota, United States, 2018.

[39] MULTOS, "multos smartcard technology," MULTOS, 2022. [Online]. Available: https://multos.com/technology/multos-smartcard-technology/. [Accessed 04 07 2023].

[40] E. B. a. G. P. N. E. Madhoun, "An Overview of the EMV Protocol and Its Security Vulnerabilities," *Fourth International Conference on Mobile and Secure Services (MobiSecServ),* pp. 1-5, 2018.

[41] shopify, "EMV: The Ultimate Guide to EMV Chip Cards," shopify, 2023. [Online]. Available: https://www.shopify.com/uk/retail/emv#6. [Accessed 13 03 2023].

[42] Visa, "Chip Card Testing & Approval," Visa Approval Services, 2021.

[44] A. Melnikovas, "Towards an Explicit Research Methodology: Adapting Research Onion Model for Futures Studies," *Journal of Futures Studies,,* vol. 23, no. 2, pp. 29-44, 2018.

APPENDICES

APPENDIX 1-TPDU

APPENDIX 1.0

ATR: 3b 6e 00 00 80 31 80 66 b0 84 0c 01 6e 01 83 00 90 00

The ATR is a message output by a contact smart card upon initial powering. It contains information about the card's properties and communication parameters.

3b: The Initial Byte, or TS, indicating the card uses direct convention.

6e: Format byte T0, indicating the number of following interface bytes.

00 00 80 31 80 66 B0 84 0C 01 6E 01 83 00: are the interface bytes, and historical bytes which provide additional details about the card.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

APPENDIX 1.1

Request 1 [30]**: TPDU: 00 a4 04 00 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 30 31 61 1c**

00 a4: select file command.

04 00: These are the parameter bytes (P1 and P2). 04 means that selection is made by name, and 00 means the first or only occurrence.

0e: length of the data field (file name), which is 14 bytes.

31 50 41 59 2e 53 59 53 2e 44 44 46 30 31: which is a string (in ASCII) represents 1PAY.SYS.DDF01. This is the name of the EMV PSE

61 1c: 61 indicates that there are still bytes available to be read, and 1c is the number of those bytes and this part is not relevant in our case.

This command represents an instruction that prompts the selection of the EMV Payment System Directory (PSE) on the card [30].

Response 1 [30]**: TPDU: 00 c0 00 00 1c 6f 1a 84 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 30 31 a5 08 88 01 01 5f 2d 02 65 6e 90 00**

00 c0 00 00 1c: representing a get response command(00c0) and the length of the expected data is 28 bytes(1c).

6f 1a: FCI Template. Which refers to the PSE.

84 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 30 31: refers to the file name 1PAY.SYS.DDF01.

a5 08: refers to FCI Proprietary Template tag and FCI Proprietary Template data length (8 bytes)

88 01 01: refers to the Short File Identifier (SFI).

5f 2d 02 65 6e: refers to language preferences 65 6e refers to the chosen language (English).

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

APPENDIX 1.2

Request 2 [30]**: TPDU: 00 b2 01 0c 00 6c 29**

00 b2: read record command.

02 0c: read record 1, SFI 1.

00: Data expected up to 256 bytes in the response.

6c 29: Are the status bytes. 6C means wrong length ,29 means the correct length (41 bytes).

Response 2 [30]: **TPDU: 00 b2 01 0c 29 70 27 61 25 4f 07 a0 00 00 00 03 10 10 50 0a 56 49 53 41 20 44 45 42 49 54 87 01 02 73 0b 9f 0a 08 00 01 05 01 00 00 00 00 90 00**

00 b2 01 0c 29: 00 b2 represents the read record command. This command is requesting to read record number (01) from the current file in the card, where the record is specified by its Short File Identifier (SFI 1). The 29 represents the expected length of the data which is 41 bytes.

70 27: 70 is tag for a record template, followed by the length 27.

61 25: 61 is a template for the directory entry, followed by its length 25.

4f 07 a0 00 00 00 03 10 10: This is the application identifier (AID)

50 0a 56 49 53 41 20 44 45 42 49 54: This is the application label 56 49 53 41 20 44 45 42 49 54 representing visa debit.

87 01 02: This is the application priority indicator with priority equals 2.

73 0b 9f 0a 08 00 01 05 01 00 00 00 00: This indicates the directory discretionary template.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

APPENDIX 1.3

Request 3 [30]**: TPDU: 00 b2 02 0c 00 6c 16**

00 b2: read record command.

02 0c: Read record 2, SFI 1.

00: Data expected up to 256 bytes in the response.

6c 16: Are the status bytes. 6C means wrong length ,16 means the correct length (22 bytes).

Response 3 [30]**: TPDU: 00 b2 02 0c 16 70 14 61 12 4f 07 a0 00 00 00 29 10 10 50 04 4c 49 4e 4b 87 01 01 90 00**

00 b2 01 0c 16: 00 b2 represents the read record command. This command is requesting to read record number (02) from the current file in the card, where the record is specified by its Short File Identifier (SFI 1). The 16 represents the expected length of the data.

70 14: 70 is tag for a record template, followed by the length 14.

61 12: 61 is a template for the directory entry, followed by its length 12.

4f 07 a0 00 00 00 29 10 10: This is the application identifier (AID)

50 04 4c 49 4e 4b: This is the application label 50 04 4c 49 4e 4b representing link.

87 01 01: This is the application priority indicator with priority equals 1.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

APPENDIX 1.4

Request 4 [30]**:TPDU: 00 b2 03 0c 00 6a 83**

00 b2: read record command.

03 0c: read record 3, SFI 1.

00: Data expected up to 256 bytes in the response.

6a 83: This indicates that the requested record does not exist.

APPENDIX 1.5

Request 5 [30]**: 00 a4 04 00 07 a0 00 00 00 03 10 10 61 2f**

00 a4: This is the select command.

04 00: 04 means that the selection is by name, and 00 means that the first or only occurrence of a file by the given name should be selected.

07: This is the length (Lc) of the command data that follows, which is 7 bytes in this case.

a0 00 00 00 03: This is RID (Registered Application Provider Identifier) for Visa.

10 10: This is Proprietary Identifier Extension (PIX) which refers to VISA credit or debit application.

61 2f: This is the response trailer for the previous command. 61 indicates that there are more data to be read with a get response command, and 2f tells the number of bytes still available.

Response 5 [30]**: 00 c0 00 00 2f 6f 2d 84 07 a0 00 00 00 03 10 10 a5 22 50 0a 56 49 53 41 20 44 45 42 49 54 87 01 02 5f 2d 02 65 6e bf 0c 0b 9f 0a 08 00 01 05 01 00 00 00 00 90 00**

00 c0 00 00 2f: This is a get response command requesting 47 bytes (2f in hexadecimal) of data from the card.

6f 2d: This FCI template which says the FCI that follows is 45 bytes (2d in hexadecimal) long.

84 07 a0 00 00 00 03 10 10: Represents the AID for the Visa credit or debit application.

a5 22: This is file control information proprietary template.

50 0a 56 49 53 41 20 44 45 42 49 54: This is application label for visa debit.

87 01 02: This is application priority indicator with priority equals 2.

5f 2d 02 65 6e: This is language preference and chosen language is English (65 6e)

bf 0c 0b 9f 0a 08 00 01 05 01 00 00 00 00: This is issuer discretionary data which indicates the application version number (AVN) of the card and the application version number data.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

APPENDIX 1.6

Request 6 [30]**: TPDU: 80 a8 00 00 02 83 00 61 14**

80 a8: This is get processing options (GPO) command. This command is used to initiate transaction processing and retrieve data necessary for transaction processing.

00 00: These are the parameters P1 and P2. P1 and P2 specify how the command data field is formatted. They are set to 00 showing that no specific requirements for command data field formatting.

02: Refers to the length which is 2 bytes**.**

83 00: This is the command data (CD) field. 00 represents no data, implying the GPO command is being issued without any specific processing options requested.

61 14: This is the status word (sw1 and sw2). 61 means that there are still additional response bytes available for the command, and 14 indicates the number of these additional bytes.

Response 6 [30]**: TPDU: 00 c0 00 00 14 77 12 82 02 38 00 94 0c 08 01 01 00 10 01 04 00 18 01 02 01 90 00**

00 c0 00 00 14: This is a get response command requesting 14 bytes of data from the card.

77 12 82 02 38 00 94 0c 08 01 01 00 10 01 04 00 18 01 02 01: This is response message template format 2 containing an application interchange profile (AIP) and an application file locator (AFL), both of which provide information necessary for the terminal to perform the transaction with the card.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

APPENDIX 1.7

Request 7 [30]**: TPDU: 00 b2 01 0c 00 6c 4b**

00 b2: Read Record command.

01 0c: Read record 1, SFI 1.

00: Data expected up to 256 bytes in the response.

6c 4b: Are the status bytes. 6C means wrong length ,4b means the correct length (75 bytes).

Response 7 [30]**: 00 b2 01 0c 4b 70 49 57 13 44 62 91 93 90 53 65 83 d2 70 52 01 61 10 00 00 00 00 0f 5f 20 16 4c 49 44 48 45 45 4d 2f 4d 41 48 4d 4f 4f 44 20 4d 20 53 2e 4d 52 9f 1f 18 30 37 36 39 32 37 35 30 30 36 37 30 30 30 30 36 31 31 30 30 30 30 30 30 90 00**

00 b2 01 0c 4b: As explained in a previous response, this is the read record command.

70 49: This is a TLV (Type-Length-Value) data object. 70 represents an EMV template. 49 is the length of the following value, in bytes.

57 13 44 62 91 93 90 53 65 83 d2 70 52 01: Track 2 Equivalent Data.

61 10 00 00 00 00 0f: This is another TLV data object. 61 refers to Application Template, 10 is the length, and the rest is the value.

5f 20 16 4c 49 44 48 45 45 4d 2f 4d 41 48 4d 4f 4f 44 20 4d 20 53 2e 4d: This is Cardholder Name: LIDHEEM/MAHMOOD M S.M

4	4	4	4	4	4	4	2	4	4	4	4	4	4	4	2	4	2	5	2	4
c	9	4	8	5	5	d	f	d	1	8	d	f	f	4	0	d	0	3	e	d
L	I	D	H	E	E	M	/	M	A	H	M	O	O	D		M		S	.	M

Table 3: Cardholder Name

52 9f 1f 18 30 37 36 39 32 37 35 30 30 36 37 30 30 30 30 36 31 31 30 30 30 30 30 30: This is Track 1 Discretionary Data.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

APPENDIX 1.8

Request 8 [30]**: TPDU: 00 b2 01 14 00 6c b7**

00 b2: Read Record command.

01 14: Read record 1, SFI 2.

00: Data expected up to 256 bytes in the response.

6c b7: Are the status bytes. 6C means wrong length, b7 means the correct length (183 bytes).

Response 8 [30]**: TPDU: 00 b2 01 14 b7 70 81 b4 9f 46 81 b0 15 aa 57 89 a8 0a f9 c8 85 eb b3 ae 8d 39 b6 77 18 bd ff 54 f3 5d 8f b0 2b 17 02 ab 11 24 d3 7c b3 44 63 af fe ee 4c 74 f9 3b 4b 83 d0 14 b6 a9 83 93 77 2f c8 77 9e 24 bc 28 cc 62 3e d4 53 80 05 f4 ab 87 f7 a3 0a 2c 4a 48 40 e5 3e 70 91 78 47 7d 9a 99 b1 01 fd a2 15 f7 0e 66 4f 50 8f 02 67 1d cb ba e2 a0 70 ad 9d 72 0c d2 97 ba cd fe 92 50 b0 bd de 45 d3 eb bf b5 e4 d9 e1 0e 03 e8 f4 36 44 c6 54 f2 1b 18 16 d2 94 55 80 cf 56 a1 0f cb ff 27 2d e2 e6 85 ab 7e 62 f0 8c 85 e9 e2 c8 3e ef ce e4 a1 fc da dc 6a 20 e2 56 97 e8 cc 90 00**

00 b2 01 14 b7: Is a read record command.

70 81 b4: This is a TLV encoded piece of data and length of data is 180 bytes.

9f 46 81 b0: ICC Public Key Certificate tag and length of data 176 bytes

15 aa 57 89 a8 0a f9 c8 85 eb b3 ae 8d 39 b6 77 18 bd ff 54 f3 5d 8f b0 2b 17 02 ab 11 24 d3 7c b3 44 63 af fe ee 4c 74 f9 3b 4b 83 d0 14 b6 a9 83 93 77 2f c8 77 9e 24 bc 28 cc 62 3e d4 53 80 05 f4 ab 87 f7 a3 0a 2c 4a 48 40 e5 3e 70 91 78 47 7d 9a 99 b1 01 fd a2 15 f7 0e 66 4f 50 8f 02 67 1d cb ba e2 a0 70 ad 9d 72 0c d2 97 ba cd fe 92 50 b0 bd de 45 d3 eb bf b5 e4 d9 e1 0e 03 e8 f4 36 44 c6 54 f2 1b 18 16 d2 94 55 80 cf 56 a1 0f cb ff 27 2d e2 e6 85 ab 7e 62 f0 8c 85 e9 e2 c8 3e ef ce e4 a1 fc da dc 6a 20 e2 56 97 e8 cc: This is ICC public key certificate.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.9

Request 9 [30]**: TPDU: 00 b2 02 14 00 6c 0c**

00 b2: Read Record command.

02 14: read record 2, SFI 2.

00: data expected up to 256 bytes in the response.

6c 0c: are the status bytes. 6C means wrong length ,0c means the correct length is (12 bytes).

Response 9 [30]**: TPDU: 00 b2 02 14 0c 70 0a 9f 47 01 03 9f 49 03 9f 37 04 90 00**

00 b2 02 14 0c: Read the 2nd record, SFI =2

70 0a: Is the template for the response message, which contains 10 bytes of data.

9f 47 01 03: Represents the public key exponent, which is the value 03.

9f 49 03 9f 37 04: DDA process includes the Unpredictable Number (9F37).

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.10

Request 10 [30]**: TPDU: 00 b2 03 14 00 6c fe**

00 b2: Read Record command.

03 14: Read record 3, SFI 2.

00: Data expected up to 256 bytes in the response.

6c fe: Are the status bytes. 6C means wrong length, fe means the correct length (254 bytes).

Response 10 [30]**: TPDU: 00 b2 03 14 fe 70 81 fb 90 81 f8 23 be 6a be 97 81 83 35 e6 aa 3a ea cd c0 af cf 50 61 55 4d 6a 17 73 62 56 bb 77 d2 84 6a f8 e5 7b 6d 20 4a a7 a9 11 f6 54 bd 4f 96 91 e7 c8 a3 83 5d d6 b0 f0 9b 9f 22 60 6d a9 11 ef 72 5a 63 9c c5 31 8a b7 be eb 9e 63 36 49 28 7a aa 6e b1 f2 37 80 24 62 b9 d9 6d 2f 6d 2f a5 8a cd 71 90 f0 6c b9 7d f8 c2 bb d0 74 db ef 87 28 70 26 1b db c0 5d 28 17 c1 5b 5a e2 de 74 34 cf b3 b7 7a db 4c 40 18 55 89 6f 78 4c 35 c7 21 fe 42 d9 89 8d 27 7c fc dc 62 7f 63 d5 3a 66 f2 bc 8f 74 2f af c9 b8 ea ea 4e 0c 36 f4 5e 9c 64 1d 24 17 e3 f3 23 7a fb 05 83 45 39 28 01 e5 25 6f 9b 1b 0c 8a a4 20 70 3e 3e a1 fa 8c f0 93 64 84 6f 83 41 a0 f9 56 01 7c a9 1c 91 c2 20 ae 37 91 75 f9 4a 6d b5 42 8f 93 79 24 86 67 73 89 12 d5 c9 33 ea 4b 20 16 4c d2 0e 0d 0b 90 00**

00 b2 03 14 fe: Is a read record command.

70 81 fb: This is a TLV encoded piece of data and length of data is 251 bytes.

90 81 f8 23 be 6a be 97 81 83 35 e6 aa 3a ea cd c0 af cf 50 61 55 4d 6a 17 73 62 56 bb 77 d2 84 6a f8 e5 7b 6d 20 4a a7 a9 11 f6 54 bd 4f 96 91 e7 c8 a3 83 5d d6 b0 f0 9b 9f 22 60 6d a9 11 ef 72 5a 63 9c c5 31 8a b7 be eb 9e 63 36 49 28 7a aa 6e b1 f2 37 80 24 62 b9 d9 6d 2f 6d 2f a5 8a cd 71 90 f0 6c b9 7d f8 c2 bb d0 74 db ef 87 28 70 26 1b db c0 5d 28 17 c1 5b 5a e2 de 74 34 cf b3 b7 7a db 4c 40 18 55 89 6f 78 4c 35 c7 21 fe 42 d9 89 8d 27 7c fc dc 62 7f 63 d5 3a 66 f2 bc 8f 74 2f af c9 b8 ea ea 4e 0c 36 f4 5e 9c 64 1d 24 17 e3 f3 23 7a fb 05 83 45 39 28 01 e5 25 6f 9b 1b 0c 8a a4 20 70 3e 3e a1 fa 8c f0 93 64 84 6f 83 41 a0

f9 56 01 7c a9 1c 91 c2 20 ae 37 91 75 f9 4a 6d b5 42 8f 93 79 24 86 67 73 89 12 d5 c9 33 ea 4b 20 16 4c d2 0e 0d 0b: Is issuer public key certificate

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.11

Request 11 [30]**: TPDU: 00 b2 04 14 00 6c 09**

00 b2: Read Record command.

04 14: Read record 4, SFI 2.

00: Data expected up to 256 bytes in the response.

6c 09: Are the status bytes. 6C means wrong length ,09 means the correct length is (09 bytes).

Response 11 [30]**: 00 b2 04 14 09 70 07 8f 01 09 9f 32 01 03 90 00**

00 b2 04 14 00 6c 09: Read Record command.

70 07: EMV proprietary template and length of data 7 bytes.

8F 01 09: Indicates that the Certification Authority Public Key Index is 09

9F 32 01 03: Indicates that the Issuer Public Key Exponent value is 03.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.12

Request 12 [31]**: TPDU: 00 b2 01 1c 00 6c 50**

00 b2: Read Record command.

01 1c: Read record 1, SFI 3.

00: Data expected up to 256 bytes in the response.

6c 50: Are the status bytes. 6C means wrong length ,50 means the correct length is (80 bytes).

Response 12 [31]**: TPDU: 00 b2 01 1c 50 70 4e 5f 25 03 23 04 01 5f 24 03 27 05 31 5a 08 44 62 91 93 90 53 65 83 5f 34 01 00 9f 07 02 ff 80 8e 10 00 00 00 00 00 00 00 00 41 03 1e 03 02 03 1f 00 9f 0d 05 b8 60 ec 08 00 9f 0e 05 00 10 00 00 00 9f 0f 05 b8 68 fc 18 00 5f 28 02 08 26 90 00**

00 b2 01 1c 50: Read Record command.

70 4e: EMV proprietary template and length of data 78 bytes.

5f 25 03 23 04 01: This is the application effective date, which is 01/04/2023.

5f 24 03 27 05 31: This is the application expiration, which is 31/05/2027.

5a 08 44 62 91 93 90 53 65 83: This is the primary account number tag 5A. The card number is 44 62 91 93 90 53 65 83.

5F 34 01 00: This indicates the PAN Sequence Number, 00 indicates that this is the first or only card associated with the account.

9F 07 02 FF 80: This is the application usage control, which indicates the issuer's restrictions on the usage and services allowed for the application. The values 'FF 80' shows these restrictions as follows.

The author converted FF 80 to binary 11111111 10000000 every bit indicates different type of restrictions in this case bit with value of 1 means no restriction and value of 0 means there is restriction.

First bit (1): Valid at terminals other than ATMs.

Second bit (1): Valid at ATMs.

Third bit (1): Valid for international services.

Fourth bit (1): Valid for domestic services.

Fifth bit (1): Valid for international goods.

Sixth bit (1): Valid for domestic goods.

Seventh bit (1): Valid for international cash transactions.

Eighth bit (1): Valid for domestic cash transactions.

Eighth bit of the second byte (1): Domestic cashback allowed.

Seventh bit of second byte (0): International cashback not allowed.

8E 10 00 00 00 00 00 00 00 00 41 03 1E 03 02 03 1F 00: It represents cardholder verification method (CVM) list.

8E 10: Emv tag and length (16 bytes).

00 00: Fail cardholder verification if this CVM is unsuccessful and Always

00 00: repeated.

00 00: repeated.

00 00: repeated.

41 03: If the transaction is for cash at an unattended terminal, plaintext PIN verification will be performed by the ICC.

1E 03: Signature If terminal supports the CVM capability.

02 03: Enciphered PIN verified online If terminal supports the CVM capability.

1F 00: Fail cardholder verification If not defined elsewhere in the list.

It means for unattended cash transactions; the terminal should use a plaintext PIN verified by the chip. If the terminal supports CVM, it can use signature verification or online PIN verification. If none of these apply, then cardholder verification should fail

9F 0D 05 B8 60 EC 08 00: IAC for Default. The value 'B8 60 EC 08 00' specifies the issuer's conditions that cause a transaction to be rejected if it could have been approved online [31].

9F 0E 05 00 10 00 00 00: IAC for Denial or rejection.

9F 0F 05 B8 68 FC 18 00: IAC for Online.

5F 28 02 08 26: Issuer Country Code, 08 26 translates to the country code 826 which is United Kingdom.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.13

Request 13 [31]**: TPDU: 00 b2 02 1c 00 6c 4b**

00 b2: Read Record command.

01 1c: Read record 2, SFI 3.

00: Data expected up to 256 bytes in the response.

6c 4b: Are the status bytes. 6C means wrong length ,4b means the correct length (75 bytes).

Response 13 [31] **: TPDU: 00 b2 02 1c 4b 70 49 8c 15 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 8d 19 8a 02 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 91 08 9f 08 02 00 a0 5f 30 02 02 01 9f 42 02 08 26 9f 44 01 02 9f 4a 01 82 90 00**

00 b2 02 1c 4b: Read Record command.

70 49: EMV proprietary template and length of data 73 bytes.

8c 15 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 8d 19 8a 02 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 8c 15: Card risk management data object list 1 (CDOL1) and length of data is 21 bytes which are a list of data objects (tag and length) to be passed to the ICC in the first GENERATE AC command [34]. The author has broken it down into following the tags:

9F 02 06: Amount, Authorised (Numeric) with length of 6 bytes.

9F 03 06: Additional associated amount with length of 6 bytes.

9F 1A 02: Terminal country code with length of 2 bytes.

95 05: Terminal Verification Results with a length of 5 bytes.

5F 2A 02: Transaction Currency Code with a length of 2 bytes.

9A 03: Transaction Date with a length of 3 bytes.

9C 01: Transaction Type with a length of 1 byte.

9F 37 04: Unpredictable Number with a length of 4 bytes.

8d 19 8a 02 9f 02 06 9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f 37 04 91 08:
Card Risk Management Data Object List 2 (CDOL2) which is list of data objects (tag and length) to be passed to the ICC in the second GENERATE AC command [34].

9F 08 02: Application version number tag with length of 2 bytes.

00 A0: Application version number.

5f 30 02: Service code with length of 2 bytes.

02 01: Service code which shows that international transactions are allowed, domestic transactions are allowed, and the transaction type is normal transactions, goods and services transactions, and cash transactions are allowed.

9F 42 02: Application currency code tag with length of 2 bytes.

08 26: Application currency code which refers to United Kingdom.

9f 44 01: Application currency exponent tag with length of 1 byte.

02: Application currency exponent equals 2.

9f 4a 01: Static data authentication tag list with length of 1 byte.

82: Static data authentication tag.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.14

Request 14 [31]**: TPDU: 00 88 00 00 04 ca 2a 88 88 61 87**

00 88 00 00: Internal authenticate command.

04: length 4 bytes.

ca 2a 88 88: It represents unpredictable number.

61 87: It means there are 135 bytes left to be retrieved and generally it will be followed by get response command.

Response 14 [31]**: TPDU: 00 c0 00 00 87 77 81 84 9f 4b 81 80 86 28 31 3f 3b d9 b9 90 48 0e 31 1c 96 a6 6f 77 d9 9a 79 99 19 bf e9 ae 85 bf fa 1a 77 95 cd 52 7d 4f 5f c3 4d 2d 98 17 84 40 43 d8 c0 75 af b5 f2 81 83 f8 ec 64 ff 19 13 41 e7 df 65 90 a0 5f 0d f9 91 98 99 34 b3**

83 5c 22 b4 6f 4b c5 6d 54 09 e5 05 d0 b1 f4 7c 14 f0 a8 e0 bd a8 3c 73 b5 fe 22 cf 8b 68 80 df e8 43 fa 30 77 67 d6 1d 2f 20 8c f7 8d 91 59 1e ed e0 d4 6f 42 e1 9d 8d 4d 90 00

00 c0 00 00 87: This is a get response command with length of 135 bytes.

77 81 84 9f 4b 81 80 86 28 31 3f 3b d9 b9 90 48 0e 31 1c 96 a6 6f 77 d9 9a 79 99 19 bf e9 ae 85 bf fa 1a 77 95 cd 52 7d 4f 5f c3 4d 2d 98 17 84 40 43 d8 c0 75 af b5 f2 81 83 f8 ec 64 ff 19 13 41 e7 df 65 90 a0 5f 0d f9 91 98 99 34 b3 83 5c 22 b4 6f 4b c5 6d 54 09 e5 05 d0 b1 f4 7c 14 f0 a8 e0 bd a8 3c 73 b5 fe 22 cf 8b 68 80 df e8 43 fa 30 77 67 d6 1d 2f 20 8c f7 8d 91 59 1e ed e0 d4 6f 42 e1 9d 8d 4d : This is the signed dynamic application data.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.15

Request 15 [31]**: TPDU: 80 ca 9f 17 00 6c 04**

80 ca: It represents a get data command.

9f 17: It represents a pin try counter.

00: It means data length is 0.

6c 04: Are the status bytes. 6C means wrong length ,04 means the correct length is (04 bytes).

Response 15 [31]**: TPDU: 80 ca 9f 17 04 9f 17 01 03 90 00**

80 ca 9f 17 04 9f 17: It represents a get data command.

01: Pin try counter 1 byte.

03: Pin try counter 3 remaining.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.16

Request 16 [31]**: TPDU: 00 20 00 80 08 24 23 07 ff ff ff ff ff 90 00**

00 20: It refers to verify command.

00 80: 80 means that the PIN is a part of the cardholder verification method.

08: Length of the data is 8 bytes.

24 23 07: It represents the plaintext PIN. In this context, the PIN used for the experiment was 2307. The 24 could represent anything.

ff ff ff ff ff: They're added to make the data meet a specific length requirement, in this case, 8 bytes.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.17

Request 17 [31]**: TPDU: 00 c0 00 00 20 77 1e 9f 27 01 80 9f 36 02 00 04 9f 26 08 03 e7 e6 aa f0 9d ca 1a 9f 10 07 06 05 12 03 a4 00 02 90 00**

00 c0 00 00 20: This is a get response command with length of 32 bytes.

77 1e: Response Message Template Format 2 with length of 30 bytes.

9f 27 01 80: It represents the Cryptogram Information Data (CID) where 9f 27 is the tag and 01 is the length and 80 is value which confirms that the transaction was approved offline.

9f 36 02 00 04: It refers to Application Transaction Counter (ATC) where 9f 36 is the tag and 02 is the length and the 00 04 is the counter value.

9f 26 08 03 e7 e6 aa f0 9d ca 1a: It refers to the Application Cryptogram (AC), or in other words TC (Transaction Certificate) where 9f 26 is the tag and 08 is the length and 03 e7 e6 aa f0 9d ca 1a is the value of the cryptogram.

9f 10 07 06 05 12 03 a4 00 02: It refers to the Issuer Application Data (IAD) where 9f 10 is the tag and 07 is the length and 06 05 12 03 a4 00 02 is the value (Application: VSDC IAD Format: 1, Key Derivation Index: 05, Cryptogram Version Number: 12 (CVN 18), Card Verification Results (CVR): A40002).

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

Appendix 1.18

Request 18 [31]**: TPDU: 80 ae 40 00 27 30 30 00 00 00 00 01 00 00 00 00 00 00 00 08 26 00 80 00 80 00 08 26 23 08 03 00 ca 2a 88 88 7b 21 5d b0 00 82 00 00 61 20**

80 ae: It refers to generate ac command.

40 00: It indicates the type of cryptogram to be generated in this case its TC (transaction approved).

27: length of data to be sent to card (39 bytes).

30 30: Authorization response code.

00 00 00 00 01 00: Amount authorized (£1.00)

00 00 00 00 00 00 00: Amount, other.

08 26: Terminal country code which is UK.

00 80 00 80 00: Terminal verification result.

08 26: Transaction currency code for the British Pound Sterling (GBP).

23 08 03: Transaction date 03/08/2023.

00: Transaction type 00 means purchase.

ca 2a 88 88: Unpredictable number.

7b 21 5d b0: Template, security environment (se).

00 82 00 00: Refers to external authenticate.

61 20: The 61 means that there are more data to be fetched, and the 20 means the number of bytes (32 bytes).

Response 18 [31]**: TPDU: 00 c0 00 00 20 77 1e 9f 27 01 40 9f 36 02 00 04 9f 26 08 fa 09 49 2d c3 cb eb b2 9f 10 07 06 05 12 03 64 00 02 90 00**

This is a get response command to get CID, ATC, AC and IAD. The complete details of the response 18 can be found in Appendix 1.18.

00 C0 00 00 20: Is used to route messages in systems based on ISO 8583 [35].

77 1E: Indicates a response message template format 2 with length of (30 bytes).

The subsequent tags are EMV tags that contain transaction information:

9F 27 01 40: Is the cryptogram information data with tag 9f 27 and length of 01 byte and value of 40.

9F 36 02 00 04: Is the application transaction counter with tag 9f 36 and length 02 bytes and value of 00 04.

9F 26 08 FA 09 49 2D C3 CB EB B2: Is the application cryptogram with tag of 9f 26 and length of 08 bytes and value of FA 09 49 2D C3 CB EB B2.

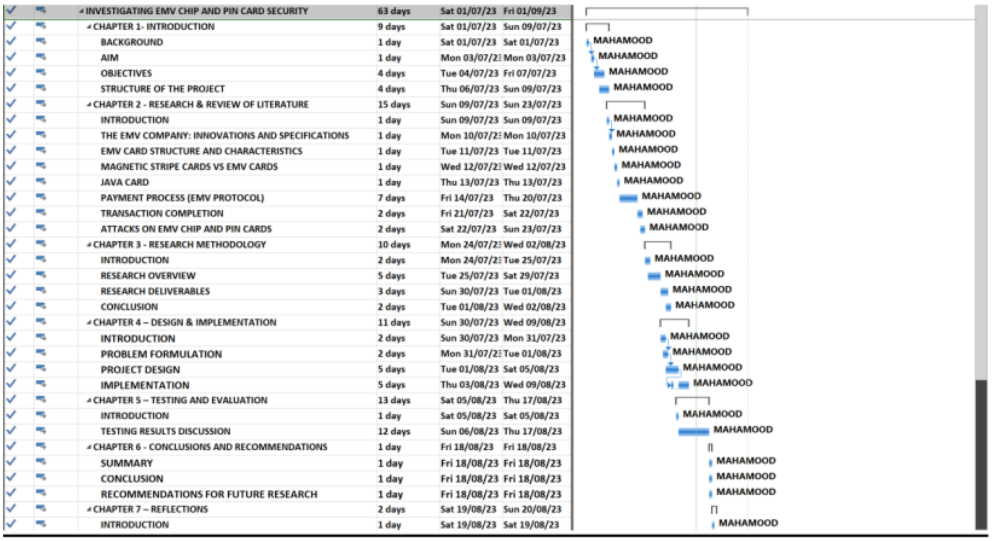
9F 10 07 06 05 12 03 64 00 02: Is the issuer application data with tag 9f 10 and length of 07 bytes and value of 06 05 12 03 64 00 02.

90 00: These are the status bytes (sw1, sw2), indicating that the previous command was processed successfully.

APPENDIX 2- PROJECT MANAGEMENT

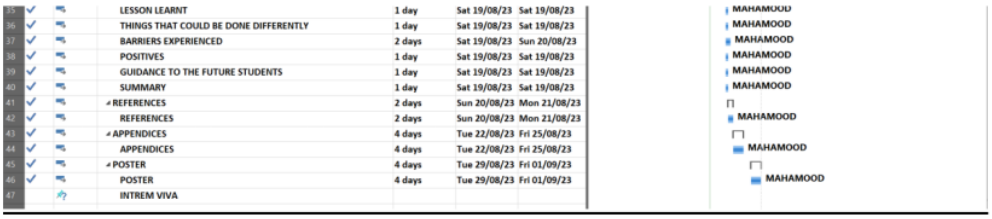
ORIGINAL GANTT CHART

The Gantt chart for the research offers a comprehensive overview of the planned project, detailing elements like task names, durations, start and end times. Although Figures (45,46) suggests that the project was initially set to finish by 01/09/2023.



	INVESTIGATING EMV CHIP AND PIN CARD SECURITY	63 days	Sat 01/07/23	Fri 01/09/23
	CHAPTER 1- INTRODUCTION	9 days	Sat 01/07/23	Sun 09/07/23
	BACKGROUND	1 day	Sat 01/07/23	Sat 01/07/23
	AIM	1 day	Mon 03/07/23	Mon 03/07/23
	OBJECTIVES	4 days	Tue 04/07/23	Fri 07/07/23
	STRUCTURE OF THE PROJECT	4 days	Thu 06/07/23	Sun 09/07/23
	CHAPTER 2 - RESEARCH & REVIEW OF LITERATURE	15 days	Sun 09/07/23	Sun 23/07/23
	INTRODUCTION	1 day	Sun 09/07/23	Sun 09/07/23
	THE EMV COMPANY: INNOVATIONS AND SPECIFICATIONS	1 day	Mon 10/07/23	Mon 10/07/23
	EMV CARD STRUCTURE AND CHARACTERISTICS	1 day	Tue 11/07/23	Tue 11/07/23
	MAGNETIC STRIPE CARDS VS EMV CARDS	1 day	Wed 12/07/23	Wed 12/07/23
	JAVA CARD	1 day	Thu 13/07/23	Thu 13/07/23
	PAYMENT PROCESS (EMV PROTOCOL)	7 days	Fri 14/07/23	Thu 20/07/23
	TRANSACTION COMPLETION	2 days	Fri 21/07/23	Sat 22/07/23
	ATTACKS ON EMV CHIP AND PIN CARDS	2 days	Sat 22/07/23	Sun 23/07/23
	CHAPTER 3 - RESEARCH METHODOLOGY	10 days	Mon 24/07/23	Wed 02/08/23
	INTRODUCTION	2 days	Mon 24/07/23	Tue 25/07/23
	RESEARCH OVERVIEW	5 days	Tue 25/07/23	Sat 29/07/23
	RESEARCH DELIVERABLES	3 days	Sun 30/07/23	Tue 01/08/23
	CONCLUSION	2 days	Tue 01/08/23	Wed 02/08/23
	CHAPTER 4 – DESIGN & IMPLEMENTATION	11 days	Sun 30/07/23	Wed 09/08/23
	INTRODUCTION	2 days	Sun 30/07/23	Mon 31/07/23
	PROBLEM FORMULATION	2 days	Mon 31/07/23	Tue 01/08/23
	PROJECT DESIGN	5 days	Tue 01/08/23	Sat 05/08/23
	IMPLEMENTATION	5 days	Thu 03/08/23	Wed 09/08/23
	CHAPTER 5 – TESTING AND EVALUATION	13 days	Sat 05/08/23	Thu 17/08/23
	INTRODUCTION	1 day	Sat 05/08/23	Sat 05/08/23
	TESTING RESULTS DISCUSSION	12 days	Sun 06/08/23	Thu 17/08/23
	CHAPTER 6 - CONCLUSIONS AND RECOMMENDATIONS	1 day	Fri 18/08/23	Fri 18/08/23
	SUMMARY	1 day	Fri 18/08/23	Fri 18/08/23
	CONCLUSION	1 day	Fri 18/08/23	Fri 18/08/23
	RECOMMENDATIONS FOR FUTURE RESEARCH	1 day	Fri 18/08/23	Fri 18/08/23
	CHAPTER 7 – REFLECTIONS	2 days	Sat 19/08/23	Sun 20/08/23
	INTRODUCTION	1 day	Sat 19/08/23	Sat 19/08/23

Figure 42: Gantt chart original 1



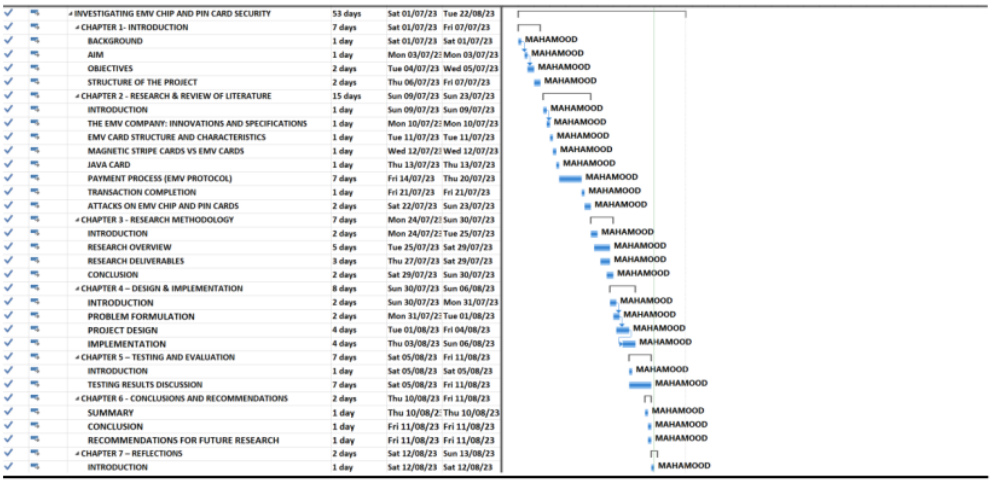
	LESSON LEARNT	1 day	Sat 19/08/23	Sat 19/08/23
	THINGS THAT COULD BE DONE DIFFERENTLY	1 day	Sat 19/08/23	Sat 19/08/23
	BARRIERS EXPERIENCED	2 days	Sat 19/08/23	Sun 20/08/23
	POSITIVES	1 day	Sat 19/08/23	Sat 19/08/23
	GUIDANCE TO THE FUTURE STUDENTS	1 day	Sat 19/08/23	Sat 19/08/23
	SUMMARY	1 day	Sat 19/08/23	Sat 19/08/23
	REFERENCES	2 days	Sun 20/08/23	Mon 21/08/23
	REFERENCES	2 days	Sun 20/08/23	Mon 21/08/23
	APPENDICES	4 days	Tue 22/08/23	Fri 25/08/23
	APPENDICES	4 days	Tue 22/08/23	Fri 25/08/23
	POSTER	4 days	Tue 29/08/23	Fri 01/09/23
	POSTER	4 days	Tue 29/08/23	Fri 01/09/23
	INTREM VIVA			

Figure 43: Gantt chart original 2

ACTUAL GANTT CHART

The Gantt chart for the research offers a comprehensive overview of the planned project, detailing elements like task names, durations, start and end times. The original Gantt chart shows that the project was initially set to finish by 01/09/2023.

However, Figures (47,48) suggest that the project will finish earlier than originally anticipated, specifically 10 days ahead of schedule (22/08/2023).



✓		INVESTIGATING EMV CHIP AND PIN CARD SECURITY	53 days	Sat 01/07/23 Tue 22/08/23
✓		CHAPTER 1- INTRODUCTION	7 days	Sat 01/07/23 Fri 07/07/23
✓		BACKGROUND	1 day	Sat 01/07/23 Sat 01/07/23
✓		AIM	1 day	Mon 03/07/23 Mon 03/07/23
✓		OBJECTIVES	2 days	Tue 04/07/23 Wed 05/07/23
✓		STRUCTURE OF THE PROJECT	2 days	Thu 06/07/23 Fri 07/07/23
✓		CHAPTER 2 - RESEARCH & REVIEW OF LITERATURE	15 days	Sun 09/07/23 Sun 23/07/23
✓		INTRODUCTION	1 day	Sun 09/07/23 Sun 09/07/23
✓		THE EMV COMPANY: INNOVATIONS AND SPECIFICATIONS	1 day	Mon 10/07/23 Mon 10/07/23
✓		EMV CARD STRUCTURE AND CHARACTERISTICS	1 day	Tue 11/07/23 Tue 11/07/23
✓		MAGNETIC STRIPE CARDS VS EMV CARDS	1 day	Wed 12/07/23 Wed 12/07/23
✓		JAVA CARD	1 day	Thu 13/07/23 Thu 13/07/23
✓		PAYMENT PROCESS (EMV PROTOCOL)	7 days	Fri 14/07/23 Thu 20/07/23
✓		TRANSACTION COMPLETION	1 day	Fri 21/07/23 Fri 21/07/23
✓		ATTACKS ON EMV CHIP AND PIN CARDS	2 days	Sat 22/07/23 Sun 23/07/23
✓		CHAPTER 3 - RESEARCH METHODOLOGY	7 days	Mon 24/07/23 Sun 30/07/23
✓		INTRODUCTION	2 days	Mon 24/07/23 Tue 25/07/23
✓		RESEARCH OVERVIEW	5 days	Tue 25/07/23 Sat 29/07/23
✓		RESEARCH DELIVERABLES	3 days	Thu 27/07/23 Sat 29/07/23
✓		CONCLUSION	2 days	Sat 29/07/23 Sun 30/07/23
✓		CHAPTER 4 – DESIGN & IMPLEMENTATION	8 days	Sun 30/07/23 Sun 06/08/23
✓		INTRODUCTION	2 days	Sun 30/07/23 Mon 31/07/23
✓		PROBLEM FORMULATION	2 days	Mon 31/07/23 Tue 01/08/23
✓		PROJECT DESIGN	4 days	Tue 01/08/23 Fri 04/08/23
✓		IMPLEMENTATION	4 days	Thu 03/08/23 Sun 06/08/23
✓		CHAPTER 5 – TESTING AND EVALUATION	7 days	Sat 05/08/23 Fri 11/08/23
✓		INTRODUCTION	1 day	Sat 05/08/23 Sat 05/08/23
✓		TESTING RESULTS DISCUSSION	7 days	Sat 05/08/23 Fri 11/08/23
✓		CHAPTER 6 - CONCLUSIONS AND RECOMMENDATIONS	2 days	Thu 10/08/23 Fri 11/08/23
✓		SUMMARY	1 day	Thu 10/08/23 Thu 10/08/23
✓		CONCLUSION	1 day	Fri 11/08/23 Fri 11/08/23
✓		RECOMMENDATIONS FOR FUTURE RESEARCH	1 day	Fri 11/08/23 Fri 11/08/23
✓		CHAPTER 7 – REFLECTIONS	2 days	Sat 12/08/23 Sun 13/08/23
✓		INTRODUCTION	1 day	Sat 12/08/23 Sat 12/08/23

Figure 44: Gantt chart actual 1



35	✓	LESSON LEARNT	1 day	Sat 12/08/23 Sat 12/08/23
36	✓	THINGS THAT COULD BE DONE DIFFERENTLY	1 day	Sat 12/08/23 Sat 12/08/23
37	✓	BARRIERS EXPERIENCED	2 days	Sat 12/08/23 Sun 13/08/23
38	✓	POSITIVES	1 day	Sun 13/08/23 Sun 13/08/23
39	✓	GUIDANCE TO THE FUTURE STUDENTS	1 day	Sun 13/08/23 Sun 13/08/23
40	✓	SUMMARY	1 day	Sun 13/08/23 Sun 13/08/23
41	✓	REFERENCES	2 days	Mon 14/08/23 Tue 15/08/23
42	✓	REFERENCES	2 days	Mon 14/08/23 Tue 15/08/23
43	✓	APPENDICES	4 days	Tue 15/08/23 Fri 18/08/23
44	✓	APPENDICES	4 days	Tue 15/08/23 Fri 18/08/23
45	✓	POSTER	4 days	Sat 19/08/23 Tue 22/08/23
46	✓	POSTER	4 days	Sat 19/08/23 Tue 22/08/23
47		INTREM VIVA		

Figure 45: Gantt chart actual 2

RISK ANALYSIS

To address potential challenges and ensure the project's success, various considerations were taken into account. The original title of the research proposal was Experimental Analysis of MITM Attack on EMV Chip and PIN Card Security. However, based on advice, the title was shortened to INVESTIGATING EMV CHIP AND PIN CARD SECURITY, though the objectives remained unchanged.

SECURITY RISKS

Conducting this experiment could raise legal concerns since it entails intercepting communications between the card and terminal. While the author used their own bank cards for the tests, if malicious individuals replicated the experiment, they might use the results to physically duplicate the card and misuse it, as highlighted in previous chapters.

It's essential to note certain limitations that could obstruct the successful completion of this experiment. These limitations primarily sourcing equipment like the Simtrace2 device, which is not only expensive but may also have extended delivery times. Additionally, obtaining a card

machine poses a challenge as they are typically reserved for business purposes and not for individual purchase. Furthermore, the sophisticated nature of the EMV protocol requiring specialized knowledge, and decoding the captured TPDU can be time-consuming.

ETHICAL FORM

See Moodle uploads (uploaded in the Deliverable Submission section with the filename Ethical Form).

dissertation final

GRADEMARK REPORT

FINAL GRADE

/100

GENERAL COMMENTS

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77

PAGE 78

PAGE 79

PAGE 80

PAGE 81

PAGE 82

PAGE 83

PAGE 84

PAGE 85

PAGE 86

PAGE 87

PAGE 88

PAGE 89

PAGE 90

PAGE 91

PAGE 92

PAGE 93

PAGE 94

PAGE 95

PAGE 96

PAGE 97

PAGE 98

PAGE 99

PAGE 100

PAGE 101

PAGE 102

PAGE 103

PAGE 104

PAGE 105

PAGE 106

PAGE 107

PAGE 108

PAGE 109

PAGE 110

PAGE 111

PAGE 112

PAGE 113

PAGE 114

PAGE 115

PAGE 116

PAGE 117

PAGE 118

PAGE 119

PAGE 120