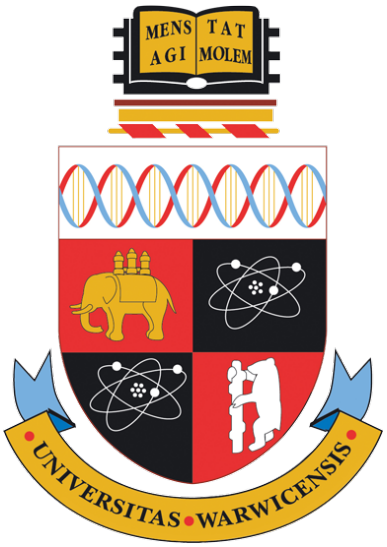
WARWICK
THE UNIVERSITY OF WARWICK



On the Security of Contactless Payment Systems

by

Mahshid Mehr Nezhad

Thesis

Submitted to The University of Warwick

in partial fulfilment of the requirements

for admission to the degree of

Doctor of Philosophy in Computer Science

Department of Computer Science

September 2023

Contents

List of Tables

List of Figures

Acknowledgments

This journey has been a personal and professional growth for me, and I would like to thank all the individuals who helped me. Firstly, I would like to thank my supervisor, Prof. Feng Hao, who guided me through my PhD. I sincerely appreciate his ongoing support and encouragement along the way and how he helped me learn to become an independent researcher. I would like to thank the Department of Computer Science and the System and Security group for the friendly environment, stimulating discussions, and all the fun breaks. I especially would like to thank my colleagues and friends, Dr Mahshid Delavar, Dr Sumanta Sarkar, Dr Shen Wang, Mohammad Nourbakhsh, and Meghdad Kurmaji.

During this research project, it was my pleasure to work with several people who also contributed to this dissertation. In Chapter 1, my sincere thanks go to Dr. Mohammed Aamir Ali, Senior Cyber Security Architect at Visa, for his valuable insights on the ecosystem of payment systems. In Chapter 3, I am thankful to Prof. Ioana Boureanu Carlson from Surrey University and Prof. Tom Chothia from Birmingham University for their input in the brainstorming of the systematization of the attacks. In Chapter 4, I'd like to thank Elliot Laidlaw, the MSc student, for helping with some experiments. In Chapter 6, I thank Dr. Maryam Mehrnezhad from Royal Holloway University of London, for helping to improve the design of the user study and Timur Yunusov, Senior Security Researcher at Payment Village, for his valuable technical feedback.

All of my work wouldn't be possible without the support of my friends and family. My deepest thanks go to my parents, my sisters, and my niece for their

constant encouragement and belief in me. A very special thanks to my husband and friend, Arash, for his unconditional support, encouragement, and patience throughout this journey. He believed in me when I doubted myself.

Finally, I'm inspired and motivated by the courageous women and men in Iran's women-life-freedom movement who fight for human and women's rights with sacrifice and resilience. This dissertation is dedicated to them with respect and solidarity.

Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy in Computer Science. It has been composed by myself and has not been submitted in any previous application for any degree.

1 Publications

Parts of this thesis have been previously published by the author in the following:

- Mahshid Mehr Nezhad and Feng Hao. "Opay: an Orientation-based Contactless Payment Solution against Passive Attacks" In Annual Computer Security Applications Conference (ACSAC), 2021
- Mahshid Mehr Nezhad, Elliot Laidlaw, and Feng Hao."Security Analysis of Mobile Point-of-sale Terminals" In 17th International Conference on Network and System Security (NSS), 2023

2 Awards

- "Best Student Paper Award" for the paper "Security Analysis of Mobile Point-of-sale Terminals" In 17th International Conference on Network and System Security (NSS), 2023.

Abstract

Contactless payment has witnessed a global surge in adoption due to heightened hygiene concerns and increased transaction limits during the COVID-19 pandemic. This dissertation offers a comprehensive analysis of contactless payment systems from four different angles including systematization and protocol analysis, attacks and vulnerabilities, countermeasures and solutions, and users' perspectives.

Firstly, we systematically explore contactless payment attacks across seven categories, categorizing them based on their objectives and the target layers within payment protocols. Vulnerabilities in these protocols are identified and mapped, exposing failures within the protocol layer. A comparative analysis of the two prominent protocols, Visa and Mastercard, is presented along with potential mitigation strategies.

Next, we analyze the security of mobile Point-of-Sale (mPoS) terminals that accept contactless transactions. Despite their convenience, they introduce potentially exploitable vulnerabilities. The findings uncover eavesdropping attacks revealing cryptographic keys in Bluetooth Low Energy (BLE) communication, man-in-the-middle (MITM) attacks tampering with mPoS terminal messages, and the risk of reverse engineering mobile phone applications to disable security features.

Subsequently, to counter mPoS-based Passive (MP) relay attacks, we propose OPay, an innovative solution based on card and reader orientation alignment. OPay demonstrates remarkable success rates, ranging from 85% to 99%, depending on the attack model, with a speedy 228-millisecond response time, meeting EMV contactless payment timing requirements. User satisfaction, measured by System Usability Scale (SUS) scores, experiences only a modest 5.28% drop, as confirmed by a user study involving 20 participants.

Finally, we bridge the gap between user perceptions of contactless payment attacks and their technical feasibility. A study involving 150 participants in the UK reveals that users accurately interpret some attacks but tend to overestimate certain risks while underestimating others. Discrepancies in the adoption of protective measures are also uncovered, despite the availability of more effective options.

Acronyms

AAC Application Authentication Cryptogram.

AAD Authenticated Application Data.

AAE Artificial Ambient Environment.

AC Application Cryptogram.

ACI Application Capabilities Information.

ADF Application Definition File.

AES Advanced Encryption Standard.

AFL Application File Locator.

AID Application Identifier.

AIP Application Interchange Profile.

AMSR Audio-jack Magnetic Stripe Reader.

APDU Application Protocol Data Unit.

API Priority Indicator.

APK Android Package Kit.

ARQC Authorization Request Cryptogram.

AT Anti Tampering.

ATC Application Transaction Counter.

ATQA Answer to Request.

ATS Answer to Select.

AUC Application Usage Control.

BLE Bluetooth Low Energy.

CA Certificate Authority.

CCC Compute Cryptographic Checksum.

CDA Combined Data Authentication.

CDCVM Consumer Device Cardholder Verification Method.

CDOL Card Risk Management Data Object List.

CID Cryptogram Information Data.

CMAC Cipher-based Message Authentication Code.

CNP Card Not Present.

CP Card Present.

CSRK Signature Key.

CTQ Card Transaction Qualifiers.

CVC Card Verification Code.

CVD Cardholder Verification Decision.

CVM Cardholder Verification Method.

DCVV Dynamic CVV.

DDA Dynamic Data Authentication.

DDOL Dynamic Data Authentication Data Object List.

DE Data Exchange.

DOL Data Object List.

DoS Denial-of-Service.

DS Data Storage.

DSDOL Data Set Definition Object List.

ECC Elliptic Curve Cryptography.

ECDH Elliptic Curve Diffie Hellman.

EDA MAC Enhanced Data Authentication MAC.

EGPO Extended GPO.

EMV Europay, MasterCard, Visa.

F2F Face to Face.

FAR False Acceptance Rate.

FCI File Control Information.

FDDA Fast Dynamic Data Authentication.

FRR False Rejection Rate.

GPO Get Processing Options.

HCI Host Controller Interface.

HTTP Hypertext Transfer Protocol.

HTTPS Hypertext Transfer Protocol Secure.

I/O Input/Output.

I2C Inter-integrated Circuit.

IAC Issuer Action Code.

IAD Issuer Application Data.

ICC Integrated Circuit Card.

IDS Integrated Data Storage.

IDSD IDS Dictionary.

IRK Identity Key.

KP Key Press.

LE Legacy Low Energy Legacy.

LL Link Layer.

LTK Long Term Key.

MAC Message Authentication Code.

MCC Merchant Category Code.

MCM Multi-Chip Module.

MITM Man-in-the-middle.

MP mPoS-based passive.

mPoS Mobile Point-of-Sale.

NED North-East-Down.

NFC Near Field Communication.

nUN numeric Unpredictable Number.

ODA Offline Data Authentication.

OOB Out-of-band.

PAN Personal Account Number.

PCII PoS Cardholder Interaction Information.

PDOL Processing Data Object List.

PIN Personal Identification Number.

PKI Public key infrastructure.

PoS Point-of-Sale.

PPSE Proximity Payment System Environment.

PR Passive Relay.

RRP Relay Resistance Protocol.

SAK Select Acknowledge.

SC Secure Connection.

SDAD Signed Dynamic Application Data.

SDS Standalone Data Storage.

SEQ Single Ease Question.

SMP Security Manager Protocol.

SPI Send POI Information.

STK Short Term Key.

SUS System Usability Scale.

TAA Terminal Action Analysis.

TK Temporary Key.

TLS Transport Layer Security.

TRM Terminal Risk Management.

TTQ Transaction Qualifiers.

TVR Terminal Verification Results.

UID Unique Identifier.

UN Unpredictable Number.

Chapter 1

Introduction

1.1 Overview

The growth of contactless payment systems has significantly impacted the financial transaction landscape, not only in the UK but also across the globe. In the UK, a notable increase of 36% in contactless payments was recorded in 2021 compared to the previous year [53]. This sharp increase is attributed to several contributing factors, for example, the upping of the contactless limit to £100, the proactive promotion of contactless payments by retailers, the enhanced accessibility offered by card acceptance devices, and the increasing comfort and familiarity that consumers have with this method of payment. Besides, the availability of different payment devices for making contactless transactions has significantly contributed to the broad adoption of contactless payments. Moreover, digital wallets such as Apple Pay [6], Google Pay [64] and Samsung Pay [114] have been key accelerators of this adoption on mobile phones and wearable devices, providing users with a more efficient and convenient way to carry out transactions. This technology became especially attractive during the COVID-19 pandemic [53].

Considering the importance of contactless payment, this dissertation studies contactless payment systems from four different angles; systematization and protocol analysis, attacks and vulnerabilities, countermeasures and solutions, and users' perspectives. It starts with a background on contactless payment protocols and closes the gap in each of the mentioned angles in each chapter. Finally, it provides a conclusion and suggests future research.

In the subsequent section, we provide an in-depth overview of the payment ecosystem. We delve into both Card Present (CP) and Card Not Present (CNP)

technologies. Then, we pivot our attention to the contactless payment systems used within CP transactions as the focus of this research scope. To wrap up, we highlight the key contributions of this research and provide an outline of the dissertation.

1.2 Payment Ecosystem

Here, we briefly discuss the various entities integral to the payment ecosystem and its associated supporting technologies. Nonetheless, our primary emphasis is on CP transactions, in particular, contactless payments, as will be discussed in Section 1.3.

1.2.1 Entities

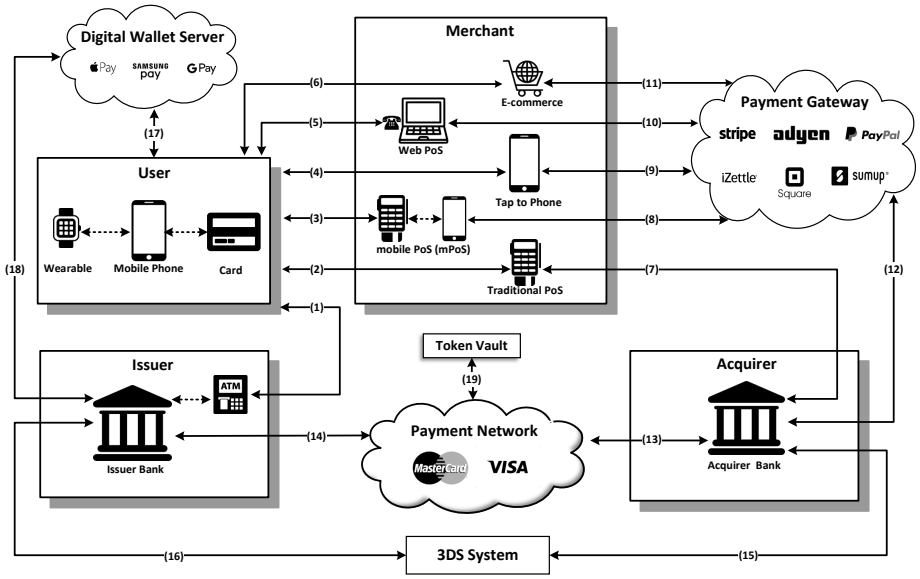


Figure 1.1: Payment System

The payment system comprises five key entities; users, merchants, acquirers, issuers, and the payment network, as depicted in Fig. 1.1.

Users

Users employ a variety of payment devices, each with different capabilities, including credit/debit cards, mobile phones, and wearable devices.

Credit/debit cards, accounting for about 50% of all UK transactions [53], can operate in either chip or contactless mode. The former relies on the chip embedded within the cards and requires the user to insert the chip card in the terminal and enter a Personal Identification Number (PIN) on the terminal pin pad, known as chip-and-PIN transactions, while the latter bypasses the need for a PIN if the transaction amount stays within a pre-defined limit (currently £100 in the UK [65]).

Mobile Phones facilitate contactless payments primarily using Near Field Communication (NFC) technology. Concurrently, the rise of QR code payments has been observed, allowing users to either scan a merchant's QR code or showcase their own to make transactions. Popular digital wallets such as Apple Pay [6], Google Pay [64], and Samsung Pay [114] accommodate not only contactless payments but, in some instances, also QR code payments [105]. To harness these platforms, users are required to enter their card information into the respective digital wallet. Most transactions necessitate authentication, commonly achieved via unlocking the mobile device. This authentication could involve methods like PIN, facial recognition, or fingerprint scan. Notably, certain digital wallets, such as Google Pay [64], allow nominal transactions without the need for phone unlocking as a form of user authentication.

Wearable devices such as smartwatches and smart jewellery, can conduct contactless transactions. These payment-enabled wearables primarily adopt NFC contactless payments and usually connect to smartphones via Bluetooth for configuration and card setup. Certain wearables (e.g., McLear Ring [93]) allow users to make contactless transactions simply by tapping on a terminal without demanding user authentication while others (e.g., Apple Watch [9]) necessitate user authentication, which might include actions like double-tapping a button.

In Chapter 6, we will explore users, as a crucial entity in this ecosystem, and delve into their perceptions of the payment ecosystem, in particular, contactless payment.

Merchants

Merchants are businesses or service providers that accept card payments for the goods or services they sell. They utilize different types of terminals to accept transactions. The nature of these transactions generally falls into two categories: Card Present (CP) and Card Not Present (CNP). CP transactions refer to the transaction between the user and the merchant during an in-person transaction where the payment device is physically present. CNP technologies refer to the communication process between the user and the merchant during remote transactions where the card is not physically present. This includes online and mobile transactions where the user uses their card information to make a purchase. Traditional Point-of-Sale (PoS), mobile PoS (mPoS), and tap-to-phone terminals accept CP transactions, while virtual terminals and e-commerce platforms accept CNP transactions, each explained below.

Traditional Point-of-Sale (PoS) terminals are usually provided by banks and establish direct communication with the acquirer bank. The acquirer bank's PoS transfers the card information directly to the bank without storing it within the third party's system. They traditionally have a stationary setup, however, wireless PoS setups are becoming more common. They accept a variety of payment options, such as chip-and-PIN, contactless, and QR-code.

Mobile PoS (mPoS) terminals are similar to traditional PoS terminals with a few differences. Firstly, they integrate a third-party payment gateway (e.g., Sumup [133], Square [124], iZettle [78]) to process transactions, with this gateway communicating with the acquirer bank. Secondly, they usually offer enhanced portability, affording both merchants and customers greater convenience. Finally, they usually work in conjunction with the merchant's mobile phone, referred to as a "merchant phone", for their management. Similar to PoS terminals, they accept various payment methods including chip-and-PIN, contactless, and QR-code transactions. A more comprehensive exploration of these terminals can be found in Chapter 4.

Tap-to-phone terminals, transform a merchant's mobile phone into an efficient payment acceptance terminal, eliminating the need for an external device, and enabling merchants to accept contactless transactions. Examples include Stripe [130], Square [126], and Ayden [10] terminal Software Development Kits (SDK) that support Mastercard's tap-to-pay method [89] on iPhone and provide a framework for application developers to integrate with their solutions [90].

Web PoS terminals, also known as virtual terminals, convert a computer

into an online terminal, making it particularly suitable for remote billing or processing credit and debit card payments over the phone. For these transactions, users are typically required to share their card details verbally. An example includes Square virtual terminal [125].

E-commerce platforms, provide click-to-pay solutions for customers through their e-commerce platforms for online transactions. These platforms enable users to make online transactions, usually by just clicking a button or going through a checkout process. Examples include Stripe [131] and PayPal [106] e-commerce systems as payment platforms.

It should be noted that *special-purpose terminals* are tailored for specific applications and might exhibit different functionalities. For example, in 2019, the Express Transit feature was introduced by platforms such as Apple Pay[7] and Samsung Pay[121]. This facilitates users in purchasing tickets via NFC-enabled mobiles and wearable devices without the need for unlocking their devices, ensuring both convenience and speed. Another example is the Pay-at-Pump system, as implemented by Visa [140] and Mastercard [88]. This integrates a payment terminal into the fuel pump itself and works by initially reserving a certain amount (typically £120) from the user's bank account at the start of fueling. Upon completion, the exact cost of the fuel dispensed is charged, and any excess reserved funds are promptly returned to the account.

Issuer

The issuer, also known as the issuing bank or card issuer, plays a pivotal role in the financial landscape by overseeing the user's account during transactions. As a financial institution, its main responsibility is to provide payment cards to consumers, a process that often involves credit evaluations and the creditworthiness of a potential cardholder. Within the scope of the payment ecosystem, the issuer is responsible for ensuring and validating whether a cardholder has enough funds or credit allowance to make a transaction. Beyond this, they employ security measures and monitoring systems to detect any irregularities or potentially fraudulent activities.

Acquirer

The acquirer, also known as the acquiring bank, is a financial institution that represents businesses in transactions, equipping them with the necessary tools to collect payments from issuers. They handle the process of retrieving money

from the issuer (via payment network) and depositing it into the business's account, facilitating the completion of the transaction. Their primary role is to interface between businesses and card networks, ensuring the successful transfer of funds.

Payment Network

Both issuer and acquirer banks rely on payment networks (e.g., Visa [142] and Mastercard [92]) to communicate with each other and facilitate the transfer of funds from the user's account (issuer) to the merchant's account (acquirer). They act as intermediaries between the banks to process the payments. In this system, when a user makes a payment, the responsibility of the payment network is to authenticate the transaction, verify that the user has sufficient funds in their account by communicating with the issuer, and send the transaction information to the issuer bank for authorization. Once the issuer bank authorizes the transaction, the payment network sends the information to the acquirer bank, which then settles the payment with the merchant [101]. These networks set terms and conditions for the transfer of funds between cardholders, merchants and their banks.

1.2.2 EMV Technologies

EMV, an acronym derived from its founding entities—Europay, Mastercard, and Visa—is a universally recognized and accepted standard for payment card and terminal operations. It has been developed to ensure interoperability and security for transactions globally. EMVCo, the organization responsible for maintaining and evolving the EMV standards, plays a pivotal role in ensuring consistent payment experiences across various regions and platforms [37]. In the section that follows, we provide an overall examination of the two primary EMV transaction types: Card Present (CP) and Card Not Present (CNP), as shown in Fig. 1.1, including the different technologies that they support.

EMV Card Present (CP)

EMVCo supports four technologies for card-present (CP) transactions: Contact Chip [35], Contactless Chip [36], Mobile [38], and QR Code [40] , as can be seen in Fig. 1.1, and denoted by numbers in parentheses below.

Contact Chip [35] is used for making payments with credit/debit cards. When making a payment using the Contact Chip, the user should perform a

chip-and-PIN transaction, which means inserting their card into a traditional PoS (2) or mPoS (3) terminal and entering their PIN to verify their identity. Additionally, chip cards can be used at the issues' ATMs (1) to withdraw cash and perform banking transactions.

Contactless Chip [36] enables in-store payments to be made using contactless chip cards without requiring physical contact with the payment terminal. To make a contactless payment, users simply tap their card on the traditional PoS (2), mPoS (3), or top-to-phone terminals (4). In addition to in-store payments, the Contactless Chip technology can also be used for contactless cash withdrawals at ATMs (1), in a tap-and-PIN transaction setting, which requires tapping the card on the ATM and inserting the PIN. This allows users to withdraw money without the need to insert their card into the machine, but instead by simply tapping their card on the ATM. This service is currently available at some issuer's ATMs, such as Barclays [15].

Mobile [38] technology enables users to make contactless payments using their NFC-enabled mobile devices. This includes mobile phones and wearable devices that support NFC technology. Similar to Contactless Chip technology, in EMV Mobile, users can make contactless transactions without the need for physical contact with the payment terminal by simply tapping their NFC-enabled device on traditional PoS (2), mPoS (3) or a tap-to-phone (4) terminal to make a contactless transaction. They potentially allow contactless cash withdrawal at ATMs (1) as well[1]. In addition to this, EMV Mobile technology specification also allows merchants to accept contactless payments on their mobile phones, the tap-to-phone technology, which provides a more flexible and cost-effective solution for accepting payments.

QR Code [40] technology enables merchants to provide payments via QR codes. QR codes are two-dimensional barcodes that can be scanned by a mobile device's camera or QR-code scanner to initiate a payment transaction. The technology supports two modes: merchant-presented mode and consumer-presented mode. In merchant-presented mode, the merchant generates a QR code that the customer scans to initiate the payment transaction. In consumer-presented mode, the customer generates the QR code that the merchant scans to initiate the transaction.

[1]Barclays initially supported this feature but closed it on June 2023 [15]

EMV Card Not Present (CNP)

Two main EMV technologies for CNP transactions are Secure Remote Commerce (SRC) [41] and 3-Domain Secure (3DS) [34], as can be seen in Fig. 1.1, and denoted by numbers in parentheses below.

Secure Remote Commerce (SRC) [41] provide a common baseline for the development of click-to-Pay e-commerce payment platforms (6). Users can make online purchases from participating merchants without having to enter their payment card information for each transaction. Instead, the payment card information is stored with the payment card issuer, with a unique token for each transaction to protect against fraud.

3-Domain Secure (3DS) [34] provides an additional security layer for online CNP transactions. It enables the exchange of data between the acquirer and the issuer to authenticate the user and approve the transaction using the 3D secure system, as shown in (15) and (16). The data exchanged during a 3DS transaction includes information about the transaction, payment method, and device being used to purchase in order to verify the legitimacy of the transaction.

EMV Tokenization - A Supporting Technology

EMV Tokenization [39] enhances both CP and CNP payments by removing the most valuable data and replacing it with a unique alternative value, called the payment token. As an example, in a contactless payment transaction using a digital wallet on a mobile phone, when a user adds a payment card to their digital wallet on an NFC-enabled smartphone, the sensitive data of the card (e.g., Primary Account Number (PAN)) should be replaced with a token that serves as a reference to the card. As it can be seen in Fig. 1.1, the payment device requests this token through the route from the Digital Wallet Server (17), to the Issuer bank (18), and to the Payment Network (14) where it has access to the Token Vault (19). This token is generated by the payment card issuer and stored in a secure Token Vault. The Token Vault is responsible for managing the life cycle of the EMV payment tokens, including issuing, revoking, and renewing them.

Back-end Authorization Flow

According to Fig. 1.1, during EMV CP transactions, the user utilizes a payment device, either by inserting it, tapping an NFC-enabled device, or scanning a QR code at the merchant's checkout via (2), (3), or (4), sometimes requiring

additional actions like PIN entry or mobile cardholder verification. Subsequently, the transaction authorization either proceeds directly to the acquirer bank (7) or is routed first through the payment gateway (8), (9) and then to the acquirer bank (12).

In the case of EMV CNP transactions, the user starts the process by providing their card information to the merchant, either on the telephone (5) or in an e-commerce platform (6). The merchant then forwards this information to the affiliated payment gateway (10), (11), which serves as an intermediary that transfers the information to the merchant's acquirer bank (12).

Regardless of the transaction type, the back-end process remains consistent. The acquirer bank communicates with the card association payment network (13) to verify the transaction's validity and check fund availability by connecting with the customer's issuing bank (14). Once the verification concludes, the issuing bank relays an approval or decline response back to the payment network (14), which then circulates the response to the acquirer bank (13). The acquirer then informs the merchant about the transaction result, and the merchant subsequently forwards this decision to the user's device. Depending on the transaction type, this transaction flow can involve the Digital Wallet Server and 3DS system as well.

For the settlement of the payment, the provider of the card reader submits a record of transactions to the corresponding payment network. Subsequently, the payment network and banks reconcile these transactions, determining the net amounts at the Bank of England. Following this process, the deduction of funds is displayed on the user's bank statement [101].

1.3 Focus: EMV Contactless Payment

Considering the vast payment ecosystem, the multiple involved entities and different EMV technologies, here, we focus on CP contactless payments that include contactless transactions. We particularly focus on tap-and-pay contactless transactions which enable users to simply tap or hold their card or NFC-enabled device close to the NFC-enabled terminal to initiate a payment.

These contactless payment systems have been targeted by numerous attacks over the years for different purposes, including creating a counterfeit replica of the contactless payment card [59], bypassing the contactless limit [17, 17, 18], bypassing the lock-screen of mobile phones for digital wallets [109, 135, 144], or other ones including relay [24, 57, 70] and pre-play [52, 61, 110] attacks.

However, the literature still lacks a thorough categorization and systematic analysis of these attacks. Such an analysis is crucial for understanding the objectives behind various attacks and recognizing the vulnerabilities in contactless payment systems that have been exploited over the years, a topic that remains under-explored.

Conversely, while mPoS terminals offer enhanced convenience for both merchants and users in the payment ecosystem, they also present inherent risks susceptible to exploitation for malicious purposes. Several attacks on these terminals have been documented over time [58, 60, 84, 97]. Even with the continuous advancements and efforts to address these vulnerabilities in payment terminals, a detailed security analysis of the latest mPoS terminals, particularly those that rely on the merchant's phone, remains absent.

Contactless payment systems have a known vulnerability to (passive) relay attacks [30, 57, 80, 85, 137]. The introduction of mPoS terminals, which integrate essential components for such attacks (NFC reader, wireless link, remote card emulator, and terminal) into one device, has amplified this risk. These terminals enable attackers to discreetly digitally pickpocket victims, termed mPoS-based passive (MP) attacks. While numerous countermeasures exist [29, 66–69, 79, 95, 117, 132, 136], they often demand changes to the usage model, sacrifice speed, or fall short against MP attacks where both parties are in close proximity. This underscores the need for solutions that prevent this attack without changing the usage model, and maintain speed, EMV compliance, and ease of use.

While these systems are tailored for end-users, there is a notable lack of insight into their perspectives on the vulnerabilities and potential attacks on these payment systems. Despite numerous user-centric studies from various countries [62, 100, 138, 147], along with others focusing on distinct aspects within the UK [3, 22, 86, 107], aiming to tackle this issue, a significant gap remains between users' perception of contactless payment attacks and the technical feasibility of contactless payment systems, particularly in the UK.

1.4 Contributions

Considering the above gaps in the literature, in this dissertation, we will examine these gaps in more detail and make the following contributions:

- Systematically examine contactless payment attacks, identifying vulner-

abilities in payment protocols and mapping them on payment protocols to understand the vulnerabilities of contactless payment protocols.

- Analyze security vulnerabilities in mobile Point-of-Sale (mPoS) terminals, highlighting significant risks in Bluetooth communication, terminal message transmission, and the merchant's mobile phone application security features.

- Propose an innovative orientation-based contactless payment solution, OPay, for MP relay attacks with a success rate of 85–99% without changing the usage model and with a minimal drop in the usability score based on the usability tests using our prototype.

- Conduct a user study with 150 participants to compare users' perceptions of contactless payment attacks with our evaluation of the technical feasibility of attacks in the literature.

1.5 Dissertation Outline

Chapter 2 reviews the key protocols for contactless payment systems, covering the protocols for card detection and EMV transaction execution.

Chapter 3 initiates with an exploration of contactless attacks. This chapter systematizes all known forms of contactless payment attacks, identifies vulnerabilities in the payment protocols, and associates these vulnerabilities with specific messages within the protocols. Here, we analyze how the EMV contactless protocols have been the target of these attacks over the years.

Chapter 4 analyzes the security of mPoS terminals. We highlight that this emergent generation of such terminals, which rely heavily on merchant's phones as a crucial component of their ecosystems, present potential security flaws and are susceptible to various attacks and vulnerabilities.

Chapter 5 introduces an innovative orientation-based contactless payment solution designed to counter MP relay attacks. This solution is based on the observation that a legitimate contactless payment transaction naturally aligns the card with the terminal surface, which can serve as a distinguishing feature between legitimate and malicious relay transactions. We have also built a prototype and have conducted user studies to evaluate its feasibility and usability.

Chapter 6 takes an empirical approach to assess the technical feasibility of the attacks discussed in Chapter 3. Following this technical feasibility assessment, we conduct a user study involving 150 UK-based participants which compares the users' concerns and perceived feasibility of attack categories with our evaluation of the technical feasibility. We also explore potential countermeasures that users can adopt to avoid potential attacks.

Chapter 7 concludes this dissertation and suggests future research.

Chapter 2

Background: Contactless Payment Protocols

2.1 Overview

Contactless payments involve distinct specifications for different parts of the payment process. For proximity cards, the ISO 14443 standard is utilized to define their electrical characteristics and govern the modulation of fields, as well as the transmission of data between the reader and the card at the lower levels of contactless payments.

In terms of the transaction flow, EMVCo has developed a comprehensive suite of books that outline a design for contactless payments, aiming to achieve universal acceptance. This set of specifications includes the "Architecture and General Requirements" (Book A), the "Entry Point Specification" (Book B), a series of kernel specifications from Kernel 2 to Kernel 8 (Books C-2 to C-8), "Security and Key Management" (Book E), and the "Level 1 Specification for Payment Systems, EMV Contactless Interface Specification".

Our discussion will first focus on ISO 14443 Protocol in Section 2.2 and then EMV Book B in Section 2.3, as they form the initial stage of the contactless transaction process. This will be followed by an examination of "Book C-3: Kernel 3 Specification (Visa)" in Section 2.4, and "Book C-2: Kernel 2 Specification (Mastercard)" in Section 2.5 which are commonly used and are the primary targets of attacks. Lastly, we'll discuss "Book C-8: Kernel 8 Specification" in Section 2.6 which signifies a critical transition from a multi-kernel architecture (C-2 to C-7) to a single-kernel architecture (C-8).

2.2 ISO 14443 Protocol

The ISO 14443 standard series outlines parameters for the identification of cards or objects in the field of contactless payment. This series of standards aims to facilitate interaction between proximity cards (such as contactless cards) and proximity coupling devices (such as card readers) by addressing various aspects like physical characteristics, power and signal interface, initialization, anti-collision, and transmission protocol.

The first part, ISO 14443-1, outlines the physical characteristics of contactless proximity cards or objects, defining their dimensions and structure and ensuring compatibility and interoperability between different devices [74].

The second part, ISO 14443-2, defines the characteristics of the radio frequency power and signal interface between proximity coupling devices and proximity cards or objects. It details the communication and power supply from the reader to the card and vice versa, typically over a range of about 10 cm [76].

The third part, ISO 14443-3, addresses initialization and anti-collision processes, enabling the reader to identify and establish communication with a specific card among multiple in the magnetic field. The initialization involves commands that activate the card. The subsequent anti-collision process aims to identify all proximity-integrated circuit cards in the field. During the initial communication setup, the reader regularly polls for proximity cards by sending Wake-UP (WUPA) command, or REQA messages. A card entering the reader's magnetic field absorbs energy from the reader, responding to WUPA or REQA messages with an Answer to Request (ATQA) message, which provides information about the card to the reader. This information, including the Unique Identifier (UID), aids in the next anti-collision phase. During this phase, the reader employs multiple Anti-collision and SELECT messages to ultimately select a single card. The selected card responds with a Select Acknowledge (SAK), signifying its compliance with ISO 14443-4, leading to the next phase of communication [75].

The fourth part, ISO 14443-4, focuses on the transmission protocol. It defines the activation and deactivation sequences of the protocol. In the active state, the reader sends a RATS command, and the card replies with an Answer to Select (ATS) response. These messages set up parameters for ensuing communications, such as limits on frame size for sending, receiving, or timing parameters. Fig. 2.1 shows the ISO 14443 protocol when there is a single card

in the field [109].

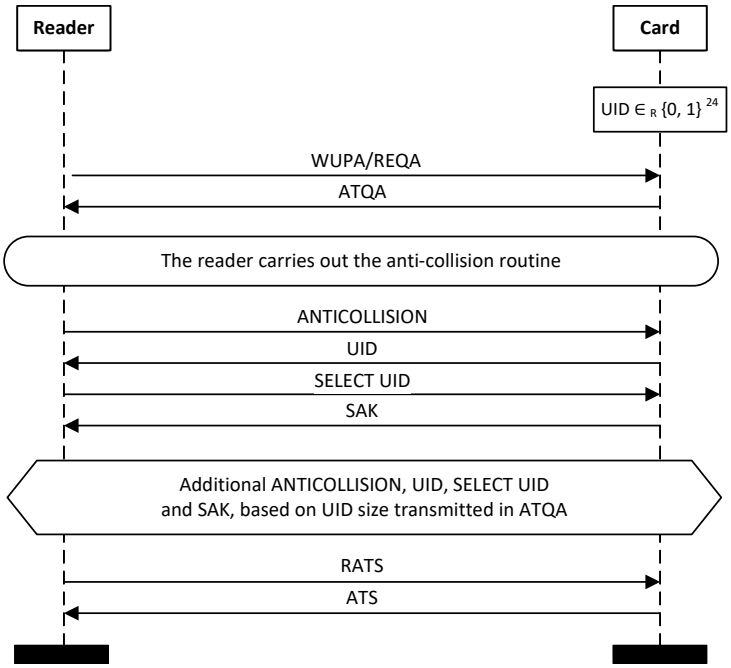


Figure 2.1: ISO14443 Protocol [109]

2.3 EMV Book B - Entry Point

As mentioned before, EMV operates on a multi-kernel (C-2 to C-7) architecture[1]. This architecture is vital for the discovery and selection of a contactless application that is mutually supported by both the reader and the card. Moreover, it facilitates the activation of the appropriate kernel for processing contactless transactions. These requirements are defined in EMV Book B - the Entry Point Specification [47].

The Entry Point has the responsibility of initiating and overseeing a sequence of interconnected functions. Each of these functions serves a specific role within the transaction flow, including Pre-Processing, Protocol Activation, Combination Selection, Kernel Activation, and Outcome Processing.

[1] the new single-kernel architecture (C-8) was proposed in 2022 and will be detailed further in section 2.6

The *Pre-Processing* is typically required for new transactions with a variable amount in an EMV mode acceptance environment. However, for transactions with a fixed amount, such as those commonly encountered in vending machines, pre-processing is usually not required as they are pre-prepared. During Pre-Processing, the Entry Point checks the configuration data. During the *Protocol Activation*, the Entry Point reads the requirements and determines the necessary actions required, such as requesting the cardholder to present a card or requesting only one card if multiple cards are detected. Next, in the *Combination Selection*, the Entry Point constructs a list of combinations mutually supported by the contactless card and the reader. If multiple combinations are supported by both, Entry Point selects the combination with the highest priority. For this purpose, the contactless card has a Proximity Payment System Environment (PPSE) that contains a list of products and applications selectable over the contactless interface. To recover the list of products and applications, as shown in Fig. 2.2, Entry Point sends a SELECT PPSE command (1). In the response, File Control Information (FCI) is provided that contains the product supported by the card, the Application Identifier (AID), and the priority of the combination (2). The priority of the combination is indicated by the Application Priority Indicator (API). Once all supported combinations have been found, and the highest priority combination has been identified, the Entry Point selects the associated card application by sending a SELECT AID command, which selects the AID with the highest priority (3). The available AIDs in the multi-kernel system include:

- Kernel 2 for Mastercard AID,
- Kernel 3 for Visa AID,
- Kernel 4 for American Express AID,
- Kernel 5 for JCB AID,
- Kernel 6 for Discover AID, and
- Kernel 7 for UnionPay AID,

In response, the card sends the FCI that contains the Processing Data Object List (PDOL), which is the list of reader-related data objects requested by the card to be transmitted in the next message, and other relevant information.Next, the appropriate kernel is activated during the *Kernel Activation* process, based

on the chosen AID. This will be elaborated upon in subsequent sections. Once the kernel completes the transaction processing, the *Outcome Processing* takes place. During this phase, the kernel delivers an outcome, such as approved, declined, and so on, upon completion of processing.

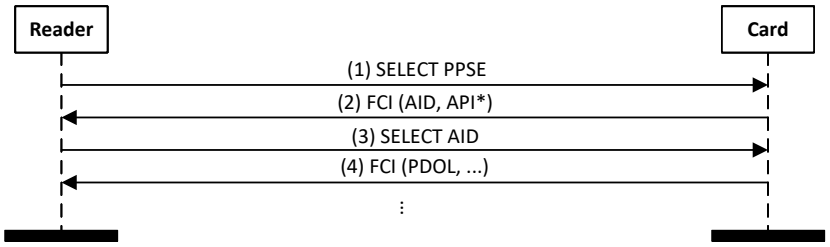


Figure 2.2: Entry Point Protocol

2.4 EMV Kernel 3 - Visa

Kernel 3 is used via Visa (PayWave) and supports a single configuration mode, EMV Mode, based on the latest specification (V2.11) [49]. Visa has removed support for the Magstripe mode in Kernel C-3 V2.6 specification [42]. In Kernel 3, the card can be presented twice; first presentment and second presentment. It also has two main functionalities; Integrated Data Storage (IDS) and Issuer Update Processing. Based on these two functionalities, the kernel analyses the data provided by Entry Point to determine whether to perform: new transaction processing only (first presentment), new transaction and IDS processing (first presentment), or Issuer Update processing (second presentment). In the following sections, we first discuss these two functionalities and then explain the transaction flow for each presentment.

2.4.1 Kernel 3 Functionalities

- **Integrated Data Storage (IDS)**: It provides a means of storing a service provider's data onto payment cards that are processed using kernel 3. An example of a service provider (referred to as an IDS Operator) is a transit system provider that uses IDS to store travel entitlement details on a payment card.

- **Issuer Update Processing:** It is an optional feature in Kernel 3 for EMV mode configurations, that allows issuers in certain terminal environments to manage card risk parameters via contactless issuer authentication or script processing. When an online authorization response contains Issuer Update Data and is supported by both card and reader, cardholders might be prompted to present their card again.

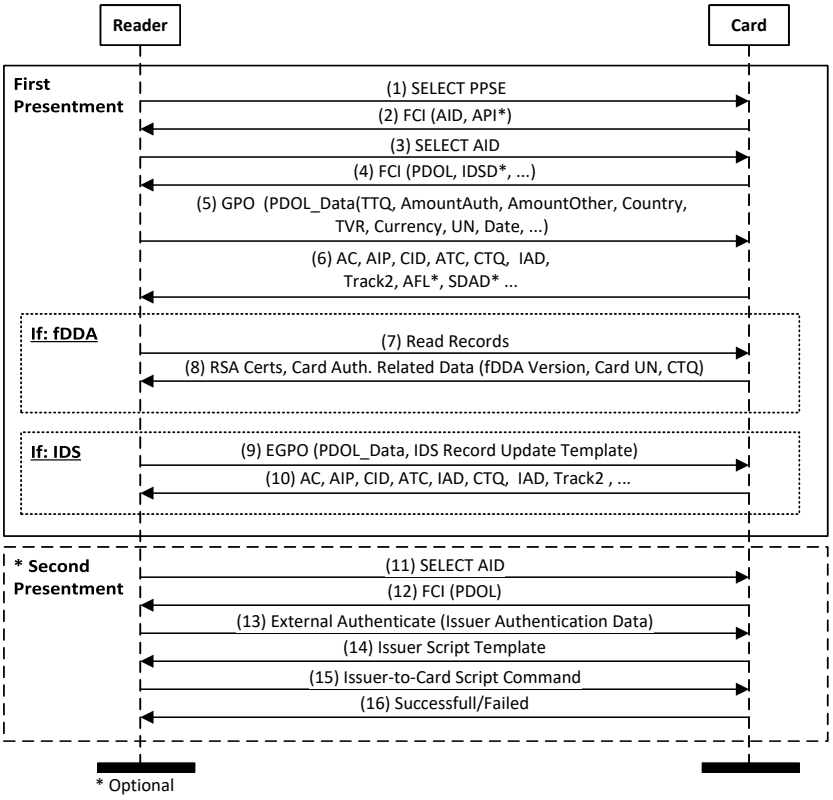


Figure 2.3: Visa Protocol based on Kernel 3 Specification

2.4.2 Kernel 3 Transaction Flow

According to Radu et. al. [109], before the start of the Kernel 3 transaction flow during the ISO 14443 protocol, Visa incorporates a relay protection protocol in its system, which operates by mandating a unique random UID, for each transaction. This UID is shared with the reader during the anti-collision process as per the ISO14443 protocol, and it is subsequently integrated into the

EMV messages within Visa's protocol. The system is designed such that this identifier must correspond at both ends, or else the transaction is identified as a potential relay attack and consequently rejected. According to [109], Visa's patent and current documentation do not elaborate on the procedure of this binding. However, it is understood that the "EMVCo NextGen" specification will detail the addition of this identifier to the Signed Dynamic Application Data (SDAD), along with the regular EMV data contained within the SDAD. If the UIDs received via the ISO14443 protocol and those within the EMV messages do not align, the transaction is dismissed as a potential relay attack. The efficacy of Visa's protection mechanism largely hinges on the complexity of assigning a specific UID, particularly in consumer devices such as mobile phones, as discussed in [109].

Kernel 3 has two presentments. In the following sections, we will discuss the transaction flow for both the first and second presentment, as shown in Fig. 2.3.

First Presentment

During the first presentment, a new transaction takes place, that can optionally include IDS as well. This initially includes running the Entry Point protocol, as explained in Section 2.3 and shown in steps (1), (2), (3), and (4) in Fig. 2.3. In message (4), the card responds with the requested PDOL, the API, as well as IDS Dictionary (IDSD) in the FCI, if IDS is to be run.

During the transaction processing, the reader sends a GET PROCESSING OPTIONS (GPO) command to the card, including PDOL Data, which is the data element requested by the card in the PDOL (5). This PDOL Data data includes Terminal Transaction Qualifiers (TTQ), the authorized amount of the transaction (AmountAuth), the other amount of the transaction (AmountOther) which is usually used for cashback transactions, the country code, Terminal Verification Results (TVR), the currency, Unpredictable Number (UN), date of the transaction, etc. TTQ indicates reader capabilities, requirements, and preferences to the card such as Cardholder Verification Method (CVM), Online options, Offline Data Authentication for Online Authorizations support[2], etc. TVR shows the status of the different functions as seen from the reader[3] [49].

The card, in turn, responds with a series of data elements which include an

[2]ODA for Online authorization is used for special purpose readers, such as transit readers.

[3]For EMV mode transactions, all of the TVR bits sent online to the acquirer shall be set to zero.

Application Cryptogram (AC), Application Interchange Profile (AIP), Cryptogram Information Data (CID), Application Transaction Counter (ATC), Issuer Application Data (IAD), Card Transaction Qualifiers (CTQ), and Track2 Equivalent Data (6). AC is the cryptogram returned by the card. AIP indicates the capabilities of the card to support specific functions in the application[4]. CID indicates the type of cryptogram (Transaction Certificate (TC), Authorisation Request Cryptogram (ARQC), or Application Authentication Cryptogram (AAC)) returned by the card and the actions to be performed by the reader. ATC is the count of the number of transactions initiated since personalization and is maintained by the application in the card. IAD contains proprietary application data for transmission to the Issuer in an online transaction. CTQ is used to indicate the card CVM requirements, issuer preferences, and card capabilities.

The Application File Locator (AFL), which points to additional data records required for the transaction, and the SDAD, a dynamic signature generated by the card, are included in the card's response for Fast Dynamic Data Authentication (fDDA) transactions. AFL is included if there are additional data records for the transaction to be returned, while SDAD is included if fDDA is supported.

FDDA: It is used if the transaction requires Offline Data Authentication (ODA), specifically Dynamic Data Authentication (DDA). DDA is an ODA mechanism that employs RSA public key cryptography and its purpose is to confirm the legitimacy of the Integrated Circuit Card (ICC). In this process, the ICC generates a digital signature on the identified ICC-resident/generated data and data received from the terminal, as defined by the Dynamic Data Authentication Data Object List (DDOL), which specifies a list of data that the card requires if the DDA method is used. In Kernel 3, ODA is implemented for readers supporting offline transactions and is performed for card-requested offline transactions. The kernel verifies the dynamic signature returned by the card and authenticates the data from the card. In this method, in addition to signing the terminal UN, fDDA also signs additional transaction dynamic data including Amount Authorised; Transaction Currency Code; and card UN.

In the transaction flow, if the fDDA is used, AFL and SDAD are sent in message (6), as well as an extra pair of command/response messages (7,8). In (7), the reader reads records in the AFL, in which the card provides RSA certificates

[4]The AIP "magnetic stripe (mag-stripe) mode is supported" bit is set to zero for products using Kernel 3 Specification Version 2.11 [49]

and data related to fDDA (8). This includes fDDA Version, Card Unpredictable Number (UN), and CTQ. Then, the dynamic signature is validated by the reader. If the validation fails, the transaction is either declined offline, sent online for authorization, or terminated, dependent on the issuer's preference.

IDS: As a result of analyzing the IDSD in the message (4), it is decoded whether any IDS Records are to be updated. The analysis of this data and the decision to update any IDS Records is IDS Operator proprietary and is outside the scope of Kernel 3 specification. If any IDS records are to be updated, the IDS Operator Application sends an instruction to the kernel to use the EXTENDED GPO (EGPO) command including the PDOL Data, along with the IDS Record Update Template (9). The card's response follows the same data structure as before in the GPO response but is updated according to the EGPO command (10).

Second Presentment

During the second presentment, only Issuer Update Processing happens. If both reader and card support Issuer Update Processing, then the cardholder can be instructed to present their card for a second time. During the second presentment, when the card is re-presented, Entry Point re-activates the kernel and the availability of Issuer Update Data indicates to the kernel that it should branch to this section. If the authorization response message contains Issuer Authentication Data and/or an Issuer Script Template, Issuer Update Processing is performed. In this optional sequence, the kernel sends a SELECT AID command to the card (11), and the card responds with the FCI, which includes the PDOL (12). Then, the reader sends an EXTERNAL AUTHENTICATE command to the card containing Issuer Authentication Data (13), to which the card responds with an Issuer Script Template (14). Using this template, the kernel forwards the Issuer-to-Card Script command to the card (15), to which the card responds indicating the success or failure of the Issuer Update Processing (16), indicating if any updates have been applied to the card.

2.5 EMV Kernel 2 - Mastercard

Kernel 2 is used via Mastercard (PayPass) based on the latest specification (V2.11) [48]. In contrast to Kernel 3 which only supports EMV Mode transactions, Kernel 2 supports two transaction modes; magstripe mode and EMV

mode. It also supports multiple functionalities including Data Storage, Optimization for Transactions without Combined Data Authentication (CDA), and Relay Resistance Protocol (RRP), which are detailed below.

2.5.1 Kernel 2 Functionalities

The following items outline the features and capabilities of Kernel 2.

- **Data Storage:** Only EMV Mode transactions support Data Storage functionality, which is an extension of the regular transaction flow. It enables the card to be used as a scratch pad or mini data store with simple write and read functionality. Two types of data storage are supported by Kernel 2: Standalone Data Storage (SDS) and Integrated Data Storage (IDS). The Kernel may support one or both data storage methods and is configured accordingly. However, the use of data storage by Kernel 2 in a given transaction is conditional on the card's indication of support for data storage, which is indicated in the response to the SELECT AID command in Application Capabilities Information (ACI).

- **Optimization for Transactions without Combined Data Authentication (CDA):** CDA is an Offline Data Authentication (ODA) method, similar to DDA, except that the card also computes a signature on the MAC. CDA includes an additional layer of security by authenticating the transaction-specific cryptogram by generating a digital signature on ICC-resident/generated data, providing a higher level of protection against certain types of fraud such as transaction tampering. In Kernel 2, when CDA is not used for a transaction, the transaction may be sped up if the public key certificates are not read from the card. In this case, the Kernel stops reading records from the card as soon as the minimum data required for the transaction is retrieved. If the card data are carefully stored in the records, then reading the public key certificates is avoided, speeding up the transaction.

- **Relay Resistance Protocol (RRP):** The Kernel supports the RRP in EMV transactions to provide protection against relay attacks based on timed Application Protocol Data Unit (APDU). The protocol relies on CDA and is included in the SDAD. If a transaction is completed without CDA, the reader cannot trust the protocol. The reader considers a transaction valid if the processing time falls within the specified window, with

an accuracy threshold and grace period. More details will be discussed in Section 2.5.2.

It is worth mentioning that Kernel 2 used to support Torn Transaction Recovery and Balance Reading features in version 2.10 [45], which were removed in the new version 2.11 released in June 2023 [48]. The former allowed the transaction data to be recovered in a torn transaction where a customer removes their card from a reader before a transaction is completed. The latter allowed the cardholder to access their current balance, enabling the user to track their spending and manage their finances.

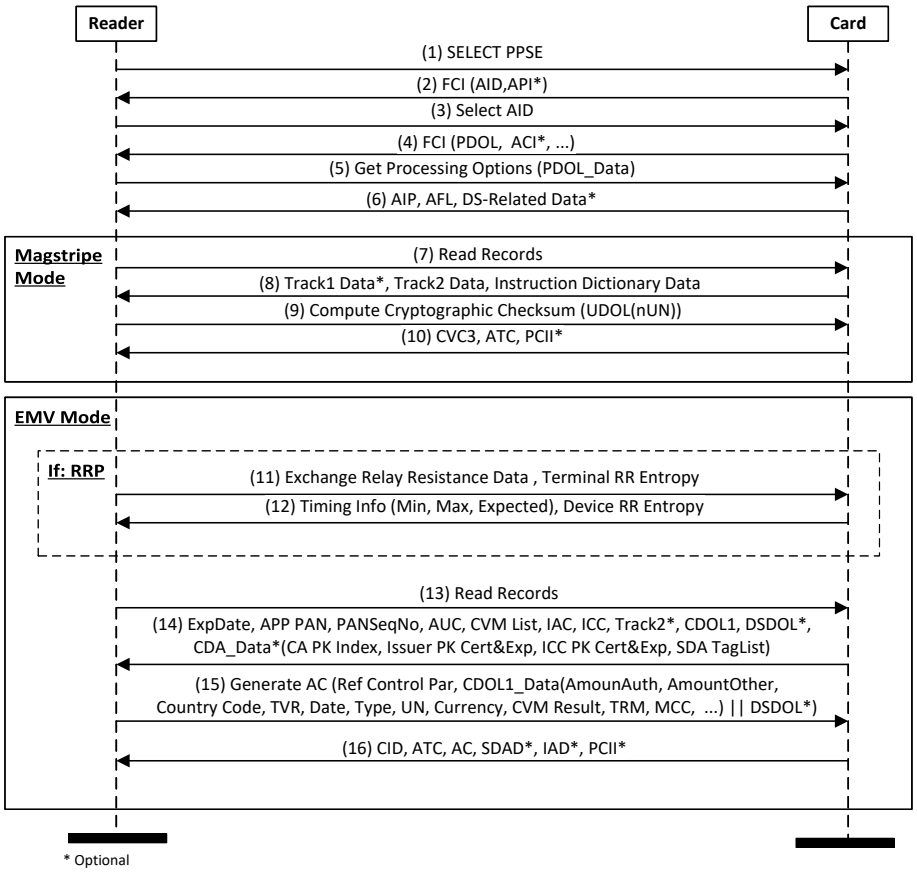


Figure 2.4: Mastercard Protocol based on Kernel 2 Specification

2.5.2 Kernel 2 Transaction Flow

The transaction flow begins with the Entry Point protocol, messages (1), (2), (3), and (4), as explained in Section 2.3. Message (4) can also include Application Capabilities Information (ACI), which indicates the support provided for SDS/IDS for the Data Storage functionality. The reader then sends a GPO command to the card with terminal-specific information coded according to PDOL (PDOL Data) (5). The card's response includes AIP, AFL, and optionally DS-related data. The AIP indicates the transaction mode, and card's capabilities with respect to functions such as cardholder verification, on-device verification, issuer authentication, support for various ODA modes, and RRP. Based on the transaction mode specified in AIP, the transaction can proceed either in magnetic stripe (mag-stripe) mode or EMV mode, as detailed below.

Mag-stripe Mode

In the mag-stripe mode, the kernel reads data records from the card (7), and the response includes Track 1 Data and Track 2 Data, as well as instructions for filling in discretionary data (8). Track 1 Data refers to data objects of track 1 and may optionally be present in the file read using the READ RECORD command during a mag-stripe mode transaction. Track 1 contains the PAN, Cardholder Name, Discretionary Data, Expiry Date, Service Code, and Field Separator. Track 2 is almost the same as Track 1, but lacks the Cardholder Name.

The kernel then sends a Compute Cryptographic Checksum (CCC) command to the Card, with a numeric UN (nUN) included in the UDOL format (9), and requests the card to return a Card Verification Codes (CVC3) cryptogram computed over the nUN. The UDOL is the DOL that specifies the data objects to be included in the data field of the Compute Cryptographic Checksum command which must include the nUN.

The Card responds to the Compute Cryptographic Checksum command including the CVC3, ATC, and PoS Cardholder Interaction Information (PCII) (10). CVC3 can include CVC3 Track1 and CVC3 Track2. CVC3 Track1 is populated if Track 1 Data is present. The PCII informs the Kernel about the indicators set in the mobile phone that may influence the action flow of the merchant and cardholder.

EMV Mode

If **RRP** is supported by both the card (indicated by AIP in the message (6)) and reader[5], the RRP will be executed. In this protocol, the Reader sends a timed C-APDU, Exchange Relay Resistance Data, to the card, which includes a random number, and Terminal Relay Resistance Entropy (11). The card responds with a random number, Device Relay Resistance Entropy, and timing estimates including the Minimum Time For Processing Relay Resistance APDU, the Maximum Time For Processing Relay Resistance APDU, and the Device Estimated Transmission Time For Relay Resistance R-APDU (12). If the reader's timings exceed the maximum limit computed, it will attempt the command processing up to two times. TVR permit the reader to be configured through the Terminal Action Codes to either decline or send transactions online if timings are outside the computed limits.

In EMV mode, after the GPO command and response (5,6) and also after the optional RRP messages (11,12), the kernel proceeds with the necessary steps for an EMV mode transaction. Firstly, it checks if the Data Set Definition Object List (DSDOL) is included in the record and if the IDS flag is set. If so, it parses the DSDOL and updates the data as required, and the terminal also updates the data accordingly.

Subsequently, the Kernel determines the method of ODA to perform and reads the data records of the card using READ RECORD commands (13). The card responds (14) and the kernel determines if CDA is supported by checking AIP. Furthermore, it verifies if CDA is supported in the Terminal Capabilities. If the CDA flag is not set, it checks if all the necessary data, including the Application Expiration Date (ExpDate), Application PAN (APP PAN), Application PAN Sequence Number (PANSeqNo), Application Usage Control (AUC), CVM List, Issuer Action Code (IAC)-Default, IAC-Denial, IAC-Online, Issuer Country Code (ICC), Track 2 Data, and Card Risk Management Data Object List 1 (CDOL1), are present in the answer to continue without CDA. IAC-Default specifies the issuer's conditions that cause a transaction to be rejected on an offline-only terminal. IAC-Denial specifies the issuer's conditions that cause the denial of a transaction without any attempt to go online. IAC-Online specifies the issuer's conditions that cause a transaction to be transmitted online on an online capable Terminal. If CDA is supported, card CDA Data Objects including the CA Public Key Index (CA PK Index), Issuer

[5]It is indicated by Kernel configuration data.

Public Key Certificate and Exponent (Issuer PK Cert&Exp), ICC Public Key Certificate and Exponent (ICC PK Cert&Exp), and Static Data Authentication Tag List (SDA Tag List) are also sent.

Following this, the Kernel requests an Application Cryptogram from the card by sending a GENERATE AC command with a Reference Control Parameter (15). This parameter is a working variable that holds the reference control parameter of the GENERATE AC command, including the AC Type and a bit indicating whether a CDA signature is requested or not. The command is accompanied by either CDOL1 Related Data or DSDOL. The CDOL1 Data comprises several fields, including Amount Authorized, Amount Other, Terminal Type, Terminal Country Code, TVR, CVM Results, Terminal Risk Management Data (TRM), Currency, Merchant Category Code (MCC), UN, ICC Dynamic Number (ICC No), and other relevant data fields.

Depending on the risk management in the card, the cryptogram returned by the card may differ from that requested in the command message. The card may return an AAC (transaction declined), an ARQC (online authorization request), or a TC (transaction approved). Subsequently, the data field in the response message to the GENERATE AC command varies depending on the usage of CDA. If CDA is not performed, the data object returned in the response message consists of CID, ATC, AC, and optionally IAD and PCII. If CDA is performed, the data object includes CID, ATC, SDAD, and optionally IAD and PCII. Finally, the Kernel performs ODA, as deemed appropriate.

2.6 Kernel 8 - Single Contactless Kernel

The C-8 Contactless Kernel Specification, also known as Kernel 8 [44], is a single kernel designed for global use in contactless payments that aims to standardize contactless kernels. This specification was developed in response to the complexity of the current multi-kernel system. Kernel 8 specification has been designed to function within the existing terminal architecture, thus preserving existing investments in terminals and equipment [50].

Kernel 8 specification aims to address privacy concerns by including a secure channel that safeguards sensitive data against eavesdropping, man-in-the-middle attacks, and relay attacks. Additionally, the specification supports Elliptic Curve Cryptography (ECC) for card authentication and biometric and mobile card verification methods [50] The specification also facilitates cloud operations, reflecting the evolving nature of payment processing. It enables

kernel processing to be split between a local terminal client and a cloud server, delegating some processing functions to the card.

2.6.1 Kernel 8 Functionalities

Similar to Kernel 2 in Section 2.5, Kernel 8 has RRP to protect against relay attacks. Additionally, it provides Privacy Protection, CVM Processing, and Data Storage features, as described below.

- **Privacy Protection:** It is a mechanism that ensures eavesdropping attacks on the communications between card and reader cannot tell the identity of the card that is used to perform the transaction. The mechanism ensures that it is impossible to tell from the payment application data if two transactions performed at the same terminal were performed by the same or different card or payment applications
- **CVM Processing:** The Kernel offers to the card in the GENERATE AC command data a list of the CVMs that it is willing to see used for the transaction and the card picks one from that list. The card informs the Kernel of its choice in response to the GENERATE AC command. The CVMs that can be used are, No CVM, Signature, CDCVM, and Online PIN.
- **Data storage:** It is an extension of the regular transaction flow such that the card can be used as a scratch pad or mini data store with simple write and read functionality.

2.6.2 Kernel 8 Transaction Flow

Kernel 8 [44] has several steps in the transaction flow as shown in Fig. 2.5 and described below.

The initialization of the transaction is similar to other kernels in steps (1), (2), (3), and (4). The processing of the transaction begins with the kernel adopting Privacy Protection and sending the GPO command to the card that contains an ephemeral ECC public key (5). Using its private key and a blinding factor, the card generates a shared secret and from that a set of session keys. In the GPO response, the card sends its blinded public key and the encrypted blinding factor so that the Kernel can compute the same shared secret and session keys. The blinding factor will permit the Kernel to authenticate the

card in conjunction with the card certificates. In addition to this card Key Data, it also sends the mandatory values of AIP, AFL, and optional values of CDOL1 and ATC (6)[6].

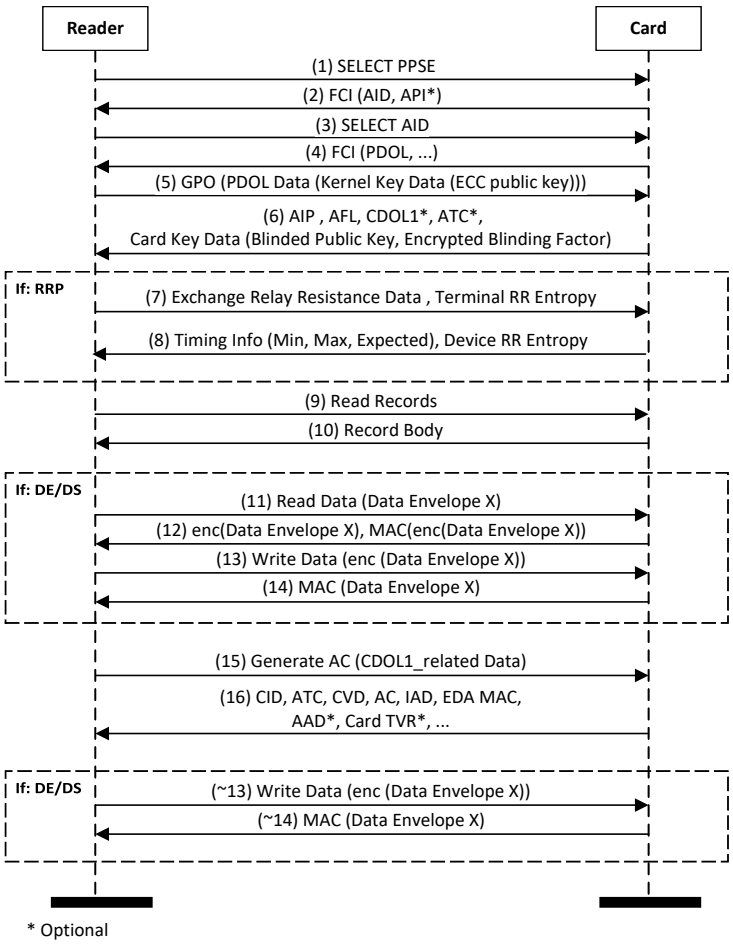


Figure 2.5: Kernel 8 Protocol based on Kernel 8 Specification

At this step, the Kernel invokes RRP if both the card and Kernel support the procedure. The command (7) and response (8) for the RRP protocol in Kernel 8 are similar to the RRP protocol discussed for Kernel 2 in Section 2.5.2, with a slight difference. In Kernel 8, the RRP is described as relying on local authentication for offline transactions and the Issuer Application Data MAC combined with online card authentication for online transactions. This

[6]GPO Response Message Data Fields can be found in Table 5.11 in the Kernel 8 specification [44].

contrasts with the Kernel 2 description where the RRP is stated to rely on CDA and the timings returned by the card are included in the SDAD.

Upon receiving the GPO response, the kernel then reads the records of the card using the Read Records command (9). The expected data field of the response message contains the record requested by the command (10).

If the Data Storage implementation option is implemented, dedicated commands (READ DATA and WRITE DATA) for explicit reading and writing of data are exchanged. When a READ DATA command is used (11), the returned data is protected by a MAC computed with one of the session keys so that the Kernel may have confidence that it was received unaltered (12). When data is written by the Kernel using WRITE DATA (13), the card returns a MAC computed over the recovered plaintext to provide confidence that it was received by the card unaltered (14). Any record data returned by the card that uniquely identifies it (for example containing the Application PAN) is encrypted by the card using one of the session keys and the AES block cipher. The kernel then performs Terminal Risk Management and Terminal Action Analysis [7].

At this point, the kernel requests an Application Cryptogram from the card by issuing a GENERATE AC command (15). The Kernel offers to the card in the GENERATE AC command data a list of the CVMs that it supports for the transaction and the card picks one from that list. The card informs the Kernel of its choice in response to the GENERATE AC command [8] as a part of CVM Processing. In the response message, the card sends the mandatory values of CID, ATC, Cardholder Verification Decision (CVD), AC, IAD, and Enhanced Data authentication MAC (EDA MAC), which is a MAC over the Application Cryptogram and IAD MAC, as well as the optional values of Authenticated Application Data (AAD)[9], and card TVR[10] (16). After the GENERATE AC command, the Kernel may send one or more WRITE DATA commands to the card, similar to messages (13) and (14).

The kernel then generates the IAD MAC (an AES-CMAC-based MAC function calculated over static card data and transaction-related data using one of the session keys) and validates the Enhanced Data Authentication MAC.

Finally, if both the card and Kernel are configured to support local authen-

[7] Details of Terminal Action Analysis is presented in section 6.4.2 of the Kernel 8 specification [44].

[8] Based on Table A.3 in Kernel 8 Specification [44].

[9] It contains Basic Encoding Rules (BER) data which may be communicated to the issuer.

[10] Terminal Verification Results returned by the Card.

tication, then the Kernel validates the card, issuer certificates, and the blinding factor.

Chapter 3

Systematization of Knowledge: Contactless Payment Attacks and Protocols' Vulnerabilities

3.1 Overview

Contactless payment systems, despite their widespread use, are prone to multi-level attacks that exploit vulnerabilities within their protocols. This chapter delivers an exhaustive systematization of these threats into seven distinct categories, each targeting a specific level: card-centric, cardholder-centric, or transaction-centric. Subsequently, the vulnerabilities within affected ISO14443, Visa, and Mastercard protocols (including the Entry Point) are mapped to provide a comprehensive overview of these systems' weak points. Our comparison of Visa and Mastercard reveals differing vulnerabilities. While both are vulnerable to card-centric attacks, Visa is more vulnerable to cardholder-centric attacks than Mastercard, while it is the opposite for transaction-centric attacks. Our observations show the root causes of these vulnerabilities, such as the vulnerable offline mode, vulnerable mag-stripe mode, unencrypted data, unauthenticated data, unauthenticated/compromised terminal, and ineffective relay protections, to highlight the historical shortcomings of these protocols. To conclude, potential countermeasures to these vulnerabilities are suggested.

3.2 Introduction

The security of contactless payment systems has been repeatedly compromised through various malicious attacks, as illustrated in Fig. 3.1. Users make contactless transactions by holding their payment device near the terminal. The close proximity of the card to the terminal initiates the transaction, enabling communication between the terminal and the payment device without physical contact, as shown in Fig. 3.1 (top). This method, often quicker than traditional contact card payments, is designed for the convenience of customers and vendors alike. However, three key points of vulnerability exist; the payment device, the terminal, and the communication link between them. Attackers can exploit these vulnerabilities in various ways, as Fig. 3.1 (bottom) illustrates where varying combinations result in different types of attacks[1].

Payment devices, such as credit/debit cards or NFC-enabled devices, are susceptible to multiple forms of compromise. A prevalent method is card cloning [52, 59, 102], wherein unauthorized copies of a card's data are created. Another example includes the installation of a malicious application on the user's NFC-enabled mobile phone [96]. Terminals, on the other hand, can be compromised in various ways. It can include physical crushing of the encryption chips as in [97], or modifications to the terminal's firmware as in [135, 144]. The security analysis of the terminals is discussed in detail in Chapter 4.

The communication link can also be compromised. In such an environment, an attacker can intercept the transaction data, using an NFC reader, or intercept and modify the transaction data, using multiple devices, such as emulators (NFC readers)[2] and proxy servers. Even though NFC communication is designed to work over a short range of a few centimetres, it can be extended over the internet, thereby facilitating attacks across different countries.

These compromises, whether used individually or in conjunction, can be exploited to perform various ways of attacks. To develop a comprehensive understanding of these attacks, in this chapter, we provide a systematization of knowledge in section 3.3 and categorize these attacks into seven attack categories based on the goal of the attacks, as well as the target of the attacks. Next, in Section 3.4, we analyze the vulnerabilities of specific EMV protocols

[1]The depiction of two emulators and one proxy in the "compromised communication" represents only one possible attack configuration.

[2]Note: NFC emulators, readers, and mobile phones capable of reading NFC signals, can be interchangeably used in attack configurations. For simplicity, we will refer to these devices as "NFC Readers" henceforth.

based on these attacks, followed by our observations in Section 3.5. Finally, we conclude this chapter in Section 3.6.

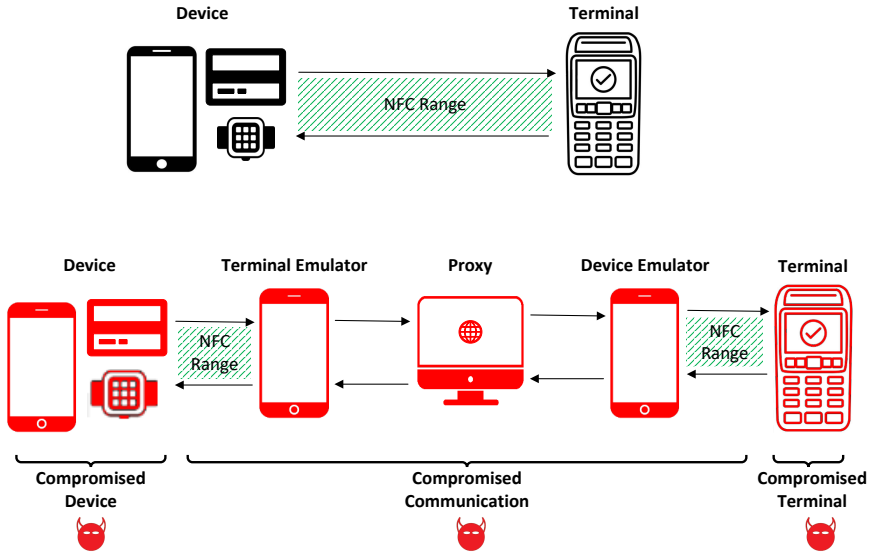


Figure 3.1: Top: A Secure Contactless Payment System. Bottom: Illustration of Potential Vulnerable Points in a Compromised Payment System.

3.3 Contactless Payment Attacks Systematization

Existing contactless payment attacks in the literature can be categorized into seven categories, each of which is detailed in Table. 3.1. Along with the categories of attacks based on their goal, they are also colour-coded based on the target of the attack in three main sections, card-centric[3], cardholder-centric, and transaction-centric. *Card-centric* attacks focus on the attacks on the ISO 14443 (Section 2.2) and EMV Entry Point (Section 2.3) protocols. *Cardholder-centric* attacks include attacks on the EMV kernel specifications (Sections 2.4 and 2.5) and focus on different methods to bypass the cardholder verification for different attack goals. Finally, *transaction-centric* attacks similarly focus on EMV kernel specifications, as well as ISO 8583 protocol, which specifies a common interface by which financial transaction card-originated messages can be interchanged between acquirers and card issuers. [73]. It should be

[3]By card, we generally mean the payment device, which can be a plastic credit/debit card or an NFC-enabled payment device.

noted that all cardholder-centric attacks can also be considered inherently transaction-centric due to their impact on the data fields that are part of the transaction details. However, it is important to distinguish cardholder-centric attacks from transaction-centric attacks. The reason for this is that some attacks specifically target the cardholder data, and these attacks are not the same as attacks that focus on the transaction details. For instance, a potential biometric-based attack on the cardholder, such as fingerprint or FaceID forgery, would not be inherently transaction-centric. This distinction implies that Table 3.1 may not definitively represent the attack classes, and a single attack might fall under two categories.

3.3.1 Data Leakage

Data leakage refers to the scenario where payment device data are obtained without authorization. This invasion of privacy is often accomplished by attackers who discreetly intercept the NFC connection. In this category of attack, one attack targets the card-centric data (Anti-collision in ISO 14443) [96], and two target transaction-centric data (Track 2 Data).

A card-centric attack reported in [96] involves the deployment of a malicious application on a user's smartphone that is designed to engage with the terminal before the legitimate card does, winning a race condition in the anti-collision process in ISO 14443 protocol in about 66% of the attempts, allowing it to track the user's contactless transactions and violate their privacy. By requesting the PDOL from the terminal, the app can retrieve transaction data, providing the attacker with specific payment information such as transaction amounts and dates.

Two other studies [33, 71] passively eavesdrop on the NFC link on the transaction data to have access to the card's data. The attack in [71] involves extracting sensitive data such as the cardholder's name and often credit card number and expiration date which are leaked in plaintext to unauthenticated NFC reader readers, Another variant of this attack [33] employs a hidden NFC reader to capture unencrypted sensitive data, which might include the card number, name, and expiry date. This permits capturing card details through NFC without the user's knowledge before complete insertion into an EMV chip-and-PIN terminal. In subsequent stages of this attack, a hidden camera is used to capture the Card Verification Value (CVV)[4].

[4]CVV guessing techniques can bypass the necessity for such camera use as reported in [4].

Attack Categories	Year	Attack Description	Affected Spec.	Affected Data	Modified	Device	Demo'ed
Data Leakage:							
Mehrnezhad et al. [96]	2016	Malicious app exploits anti-collision to request data	ISO14443	Anti-collision	×	Phone	✓
Heydt et al. [71]	2007	Steal card details via hidden NFC reader	All	Track 1/2 (Name, PAN, EXPDate)	×	Card	✓
Emms & Moorsel [33]	2011	Steal card details via hidden NFC reader and a camera	All	Track 1/2 (Name, PAN, EXPDate, CVV)	×	Card	✓
Relay:							
Radu et al. [109]	2022	Visa relay protection protocol bypass by setting UID	ISO14443	UID	✓	Phone	✓
Radu et al. [109]	2022	Mastercard Relay Protection Protocol (RRP) timing bypass in specific position	Mastercard	RRP Timing	×	Phone	✓
Multiple [20, 24, 30, 57, 70, 80]	All Time	Relay payment information between a card and a distant terminal	All	-	×	Card	✓
Pre-play:							
Roland & Langer [110]	2013	Downgrade to mag-stripe mode and pre-play attack	Mastercard	AIP, nUN, CVC, ATC	✓	Card	✓
Fillmore [52]	2015	Downgrade to mag-stripe mode and transaction clone	Mastercard	AIP, nUN, CVC, ATC	✓	Card	✓
Galloway & Yunusov [61]	2019	Sending predictable UN via a compromised terminals	Mastercard, Visa	UN, ATC	✓	Card, Phone	✓
Counterfeit Card Replica:							
Paget [102]	2012	Extract card numbers via NFC reader to clone mag-stripe cards	Visa, Mastercard	Track 1/2 (PAN, EXPDate)	×	Card	✓
Fillmore [52]	2015	Downgrade to mag-stripe and exploit dCVV vulnerabilities	Visa	Track2, AIP	×	Card	✓
Galloway [59]	2020	Read data from both mag-stripe and EMV modes and substitute them	Visa, Mastercard	Track 1/2	×	Card	✓
Limit Bypass:							
Basin et al. [17]	2021	Card Brand Mixup: change Mastercard to Visa to bypass Mastercard PIN	Entry Point (Mastercard)	AID	✓	Card	✓
Galloway & Yunusov [61]	2019	Manipulating CDCVM and CVM to bypass cardholder verification	Visa	TTQ (CVM), CTQ (CDCVM)	✓	Card, Phone	✓
Basin et al. [16]	2021	Manipulating CDCVM to bypass cardholder verification	Visa	CTQ (CDCVM)	✓	Card	✓
Basin et al. [18]	2023	Exploiting offline card validations and bypassing CVM	Mastercard	CVMList, IAC, CA PK Index, AID	✓	Card	✓
Emms et al. [31]	2013	Offline PIN verification wirelessly	Visa	PIN	×	Card	✓
Emms et al. [32]	2014	Unlimited value transactions when in foreign currency	Visa	Currency	✓	Card	×
Lock-screen Bypass:							
Yunusov [144] and Radu et al. [109]	2021, 2022	ApplePay-Visa: sending "magic string" and setting ODA for Online Authorization	Visa	TTQ (ODA), CTQ (CDCVM) Magic String	✓	Phone	✓
Yunusov et al. [135]	2021	GooglePay-Visa: set CVM to zero	Visa	TTQ (CVM)	✓	Phone	✓
Yunusov et al. [135]	2021	GooglePay-AmericanExpress: set CVMRequired to zero	AmericanExpress	CVMRequired	✓	Phone	✓
Yunusov et al. [135, 145]	2021, 2022	GooglePay-Mastercard: downgrade and clome transaction using [110]	Mastercard	AIP, CCC, nUN, CVC3, ATC	✓	Phone	✓
Yunusov et al. [135, 144]	2021	SamsungPay-Visa V.1: modify the AmountAuth to bypass only zero-value req. of SamsungPay	Visa, ISO8583	Visa: AmountAuth, ISO8583: AmountBill	✓	Phone	✓
Radu et al. [109]	2022	SamsungPay-Visa V.2: modify the AmounOther as it is not checked for zero-value req.	Visa	AmountOther	✓	Phone	×
Yunusov [144]	2021	SamsungPay-Mastercard V.1: Card Brand Mixup	Mastercard	AID, CID	✓	Phone	✓
Yunusov et al. [135]	2021	SamsungPay-Mastercard V.2: Compromised Terminal	Mastercard	AmountAuth, MCC	✓	Phone	✓
Yunusov et al. [135, 144]	2021	SamsungPay-AmericanExpress	AmericanExpress	AmountAuth, MCC	✓	Phone	✓
Yunusov et al. [135, 144]	2021	ApplePay-Mastercard	Mastercard	AmountAuth, MCC	✓	Phone	✓
Cryptogram Exploitation:							
Yunusov et al. [135]	2021, 2022	Cryptogram Confusion: change the declined type (AAC) to an authorized one (ARQC)	Visa	CID	✓	Card, Phone	✓
Yunusov et al. [146]	2021, 2022	Modifying APDU response to bypass card's functionalities and then Cryptogram Confusion attack	Visa	CID, APDU Response, ATC	✓	Card, Phone	✓
Chothia et al. [24]	2015	Sending unauthenticated corrupted AC as it is not included in SDAD	Visa	AC, SDAD	×	Card	×
Basin et al. [16]	2021	sending unauthenticated cryptogram in offline transactions	Visa, Old Mastercard	AC	×	Card	×

Table 3.1: EMV Contactless Attacks Systematization

Other tools such as NFC Reader [115] and Pro Credit Card Reader [98] can also be used for capturing the leaked data. We have used these tools to wirelessly capture the data from a contactless credit/debit card. Specifically, the card's Track 2 Data, which contains the primary account number and expiration date, is collected. The card's transaction history (if any) can also be captured. Results can be found in Appendix A.1.

3.3.2 Relay

Malicious actors may exploit NFC technology's extended range to execute relay attacks, intercepting payment information between a card and a distant terminal. This normally involves placing an NFC reader (terminal emulator) close to a victim's contactless payment device in order to initiate a contactless transaction and then transmitting the transaction data wirelessly to another remote NFC reader (card emulator), which is placed close to a legitimate terminal, to complete the transaction. This involves two devices: the first captures the payment data, transmitting it to the second, closer to the terminal, which relays the information to allow unauthorized real-time transactions unbeknownst to legitimate users.

Attackers employ various devices to relay transaction-centric data, including Android devices as suggested by prior studies [30, 57, 80, 85, 137], and customized hardware and software [70], such as the NFCMiTM tool [128], which uses two PN532 readers and a Raspberry PI. Attackers may combine these devices to enhance attack efficacy, as demonstrated by [20, 24]. To maintain practicality, relay attacks must adhere to the EMV requirement of a relay time within 500 milliseconds [46]. Some techniques, like those in [24], operate within this limit, ensuring compatibility with EMV systems.

Our successful replication of the relay attack on a Visa contactless card, using open-source tools as used in [109], demonstrates that such attacks still remain possible, highlighting the continued vulnerability of contactless payment systems. Detailed data logs associated with this experiment are available in Appendix A.2.

Moreover, it is worth mentioning that recent relay attacks reported in [94] conveniently combine the necessary equipment into a single compact device, mPoS terminals facilitating attacks by reaching the victim's pocket or purse for digital pick-pocketing. This form of attack is called mPoS-based passive (MP) relay attack and will be discussed in detail in Chapter 5.

As discussed in Sections 2.4 and 2.5, Visa and Mastercard have their own relay protection protocols. However, researchers have found ways to bypass these protocols as well, making them still vulnerable to relay attacks. Researchers in [109] bypassed Visa's relay protection protocol by taking advantage of the ability to manually set the UID on a rooted Android device during the ISO 14443 protocol, which is considered card-centric data. This allows the attacker's device to impersonate the real card during a transaction by producing the same UID as the real card. Once the attacker obtains the UID from a legitimate card, they set it as their phone's UID. The lack of a roundtrip timing measurement within Visa's protocol and the absence of distance bounding give attackers time to perform this UID manipulation and proceed with a regular transaction.

Mastercard's Relay Protection Protocol (RRP) is also bypassed in [109] when used in different positions and angles, and is considered a transaction-centric attack. The attack exploits the variable response times of payment cards based on their distance from the reader. This makes it challenging for the reader to distinguish between a relayed card at an optimal position and a legitimate card at a less ideal position. This difference in response times can be utilized to perform a relay attack, even with standard hardware. RRP timings are considered transaction-related data.

3.3.3 Pre-play

Pre-play attacks, also known as transaction cloning, involve pre-recording transaction-related data for future utilization. An example of such an attack was first presented in 2013 [110], where researchers exploited a card field vulnerability and executed a combined pre-play and downgrade attack on Mastercard's magstripe mode by manipulating unauthenticated static data to downgrade to mag-stripe mode via modifying AIP. This attack exploits the low entropy of the nUN in the mag-stripe mode and pre-calculates CVC from a genuine contactless mag-stripe card and stores them on a functional card clone for later use. Ideally, such transactions should be rejected according to ATC, but not only the issuers do not check ATC values, but the researchers also make sure not to use ATCs lower than the previous values. Similarly, Filmore [52] clones Mastercard transactions by reading and copying card records, generating a dictionary of responses for all possible terminal random numbers, downgrading to mag-stripe, replaying the stored records to the terminal, and querying the dictionary for the returned nUN provided by the terminal.

Later in 2019, researchers in [61] showed these attacks are still feasible, by exploiting vulnerabilities in key generation, UN, and ATC in contactless transactions. They demonstrate that compromising a terminal and sending a predictable UN is feasible. In this attack, the attacker can read information from a card as well as Android wallets, on the NFC interface, and replay it on the compromised terminal that always sends a predictable UN. This attack suggests that on both Visa and Mastercard, there are no limitations on repeating similar UNs to the card or on the ATC value enforced by issuers.

3.3.4 Counterfeit Card Replica

A counterfeit card replica attack involves fraudsters creating fake mag-stripe copies of legitimate payment cards to carry out unauthorized transactions. In [102], fraudsters utilize NFC readers to intercept and extract transaction-centric data, specifically Track 1 and Track 2, looking for card numbers and then cloning them onto a blank functional mag-stripe card for fraudulent transactions.

Another example is demonstrated in [52], with a focus on cloning Visa cards on the Dynamic CVV (dCVV) mode, a legacy mag-stripe equivalent mode at the time, which was found to be significantly flawed. A key vulnerability of the dCVV algorithm was its lack of a random number as an input. The algorithm used the ATC and the PAN, but not the UN, which made cloning quite straightforward. To exploit this vulnerability, attackers could read and copy the card records, activate the mag-stripe bit in the AIP, and replay these stored records to the terminal.

Galloway [59] demonstrated in 2020 that mag-stripe card cloning still remains feasible. The initial step involves reading data from both the EMV and mag-stripe interfaces within the Track 1 and Track 2 data. Once this data is collected, a comparison is undertaken to substitute the differences and similarities between the two sets of data. A crucial part of this process is the determination of the card security code value for each interface. After identifying these security codes, they are substituted into the mag-stripe tracks, effectively allowing for the creation of a new card using the full information harvested from Track 1 and Track 2 data.

The replicated mag-stripe cards would require a terminal that supports this mode of transaction. Hence, they can be used by either the fallback method or with a dedicated mag-stripe interface. Fallback is a process that occurs when the cards fail to be read by the chip-inserted method. This can be achieved

by covering the chip with tape or not fully inserting the card. This should be repeated several times until the terminal prompts for the card to be swiped [59].

3.3.5 Limit Bypass

A contactless limit bypass attack allows unauthorized users to exceed the transaction limits on contactless payment devices. For instance, the UK has a limit of £100 for contactless transactions [65], however, attackers have found ways to bypass this limit without requiring any cardholder verification method.

Basin et al. [17] bypass the contactless limit on Mastercard cards by changing the card's AID, a card-centric attack, from a Mastercard to a Visa. In this scenario, which is called the Card Brand Mixup attack, the AID of a Mastercard is altered to resemble a Visa card, allowing the attacker to utilize the attack method in [61] and [16] which exploits the cardholder related data to bypass the contactless limit. In [61], they bypass contactless payment limits on the Visa by compromising the cardholder-centric data and manipulating the Consumer Device CVM (CDCVM). They set the CVM to zero in the TTQ, indicating that the card does not require cardholder verification, and set the CDCVM to one in the CTQ, informing the terminal that cardholder verification has already been performed on the device. Basin et al. [16] demonstrated a similar attack on Visa cards as well, by only setting the CDCVM bit in CTQ.

Another cardholder-related attack (which also contains some card-related and transaction-related data) is reported in [18]. Following detection and patching efforts of the Card Brand Mixup attack in [17], an alternative bypass method for Mastercard cards was proposed that exploits vulnerabilities in payment terminals during offline card validation using a Public Key Infrastructure (PKI). This included replacing the CA Public Key Index with an invalid one, which allows the terminal's PKI checks to be bypassed, and then either downgrading (to a paper signature) or removing the CVM List. Finally, they clear the Issuer Action Code (IAC)-Denial by replacing it with all zeroes to avoid a declined transaction. The reason to do this is to avoid offline declined transactions since IAC, Terminal Action Code (TAC), and Terminal Verification Results (TVR) are inputs to make a decision for the transaction. The formula "(IAC-Denial OR TAC-Denial) AND TVR" should always equal zero so that it can either accept offline or request online authorization. This attack can also involve altering the AID of Mastero cards with the Mastercard AID which is only

applied to Maestro cards and tricking the terminal into carrying out the default Mastercard transaction flow. However, it has been mentioned that Mastero cards are proprietary and thus unavailable to them, so this part of the attack is not demonstrated.

An old vulnerability on Visa cards that affects cardholder-related data is reported in [31] in 2013 where the PIN was transmitted wirelessly which enabled potential interception and guessing. They discovered that many UK contactless cards allowed for offline verification at the time and that there was a subset of commonly used PINs, making them susceptible to guessing (e.g., reading other cards could increase the chance of guessing a PIN). This flaw has since been addressed, and the option to verify the PIN through contactless methods is no longer available.

Researchers in [32] had also found ways to bypass the contactless limit on visa cards by exploiting the "Currency" value in the transaction-centric data. They reported that contactless credit and debit cards approved unlimited value transactions when the transaction was carried out in a foreign currency without requesting the PIN, and without requesting that the PoS terminal go online to perform additional checks. This attack has not been demonstrated.

3.3.6 Lock-screen Bypass

As referenced in Section 1.2.1, both NFC-enabled mobile phones and wearable devices equipped with NFC can facilitate contactless payments. These devices typically necessitate a cardholder verification method, such as a lock-screen featuring Face-ID, PIN, or fingerprint authentication. However, in this specific type of attack, the lock-screen security measure can be circumvented, enabling attackers to execute contactless transactions without needing to unlock the mobile phone[5].

The Express Transit feature, as discussed in Section 1.2.1 on special purpose readers, represents one method that can be exploited to perform this particular type of attack that targets the security features of various systems. For instance, Apple Pay, which is available in 14 different countries across Asia-Pacific, Europe, and North America [8], utilizes a so-called "magic string", which is a specific byte sequence originating from transit readers. In contrast, Samsung Pay, that its transport mode is only advertised to function in Transport for London (TFL) [122], triggers ticket charges through zero-value transactions while the

[5]This type of attack has only been reported in relation to NFC-enabled mobile phones.

device remains locked [109]. Google Pay, on the other hand, authorizes certain transactions of minimal value without requiring user authentication and does not feature a dedicated transport mode [63]. In terms of payment networks, Mastercard implements a check for the Merchant Category Code (MCC) which must fall within the transit range as outlined in [12].

Researchers have found various ways to bypass each feature [109, 135, 144] for each combination of digital wallets (Apple Pay [6], Google Pay [64], Samsung Pay [114]) and card brands (Visa [142], Mastercard [92], American Express [51]). In this category, three attacks bypass the lock-screen by targeting cardholder-centric data (GooglePay-Visa and GooglePay-American Express, ApplePay-Visa) while the rest of the attack combinations target transaction-centric data.

For ApplePay-Visa, Yunusov [144] and Radu et al. [109] demonstrated the attack by first sending a "magic string" to the victim's device to convince it that it is communicating with a transit terminal, then bypassing the lock-screen by setting the "Offline Data Authentication (ODA) for Online Authorizations supported" bit in the TTQ which is used for special purpose readers as outlined in Section 2.4.2. Researchers in [109] explored the possibility of executing this attack for over-the-limit transactions by modifying CDCVM in CTQ, as well as the copy of CTQ, enabling the circumvention of contactless limits. This highlights the severity of the attack, including the ability to bypass the lock screen for Apple Pay and Visa transactions with no transaction amount restriction. The study conducted a test transaction of £1000 to demonstrate the exploit.

For Google Pay digital wallets, Yunosov et al. [135, 145] have bypassed the lock-screen of mobile phones for Visa, American Express, and Mastercard. For GooglePay-Visa, they manipulate the TTQ field to set the CVM to zero, indicating that no CVM is required for the transaction, in a condition that the phone screen is active, which is not considered a difficult task. For GooglePay-AmericanExpress, they used a modified PoS to initiate a payment, then employed a Man-in-the-middle (MITM) attack for the American Express protocol to alter the CVM requirement bit to zero. For GooglePay-Mastercard, they first perform a downgrade attack to mag-stripe and change the AIP to convince the terminal that the wallet does not support any mode except mag-stripe mode, as explained in [110], and then create a clone of Mastercard transactions on GooglePay. Then, they bypass the unlock requirement by changing a bit (called CVMResults) in the Compute Cryptographic Checksum (CCC) that

indicates the phone should be unlocked due to the high-value amounts. Due to the low entropy of the unpredictable number (nUN) which allows CVC values to be pre-calculated, and the ATC values that are out of order, a successful clone of GooglePay-MasterCard transactions can be made.

Researchers have also found ways to bypass the SamsunPay lock-screen for different card brands, which would require bypassing the Samsung Pay zero-value requirement. For SamsungPay-Visa, in the first attacks, Yunusov et al [144] [135] bypass the lock-screen by first initiating a £1.00 payment with a modified PoS system and leverage a MITM attack to change the amount field in the Generate AC command from £1.00 to £0.00. This action allows the cryptogram to be valid only for £0.00. The next step involves making an ISO 8583 Authorization request with the modified cardholder Billing Amount field (£1) to charge the user and get the cryptogram accepted.

In the second attempt, Radu et al. [109] report the possibility of a SamsungPay-Visa lock-screen bypass attack for cashback transactions. They propose initiating a transaction with some value in the "AmountOther" field which is intended to be used in cashback transactions, and keeping the "AmountAuth" value zero, to meet the zero-value requirement of SamsungPay. It is claimed that the zero-value transaction requirement is only applied to the "AmountAuth". However, this attack has not been demonstrated.

For Samsungpay-Mastercard, Yunusuv et al. [135] proposed two variants of the attack. They first changed the Mastercard card to a Visa using the Card Brand Mixup attack by changing the AID as in [17], requested a cryptogram from the locked phone, and completed the attack by executing the Cryptogram Confusion attack [144], which is changing the Cryptogram Information data (CID) from a failed cryptogram type to a successful type and will be explained in Section 3.3.7. After the vulnerability reported in [17] was patched, they suggested another variant involving the initiation of payment for £1.00 with a compromised PoS and a subsequent MITM attack to change the amount field in the Generate AC command from £1.00 to £0.00. They also changed the MCC to a transit operator code (4111), as Mastercard only works within the transport scheme range. SamsungPay-AmericanExpress and ApplePay-Mastercard attacks are also claimed to work similarly to SamsungPay-Mastercard.

3.3.7 Cryptogram Exploitation

This category of attacks targets cryptograms, either by changing their type or sending an unauthenticated one, targeting transaction-centric data. In the former attack, which is called the Cryptogram Confusion attack, Yunusov et al. [135] [146] modified the CID type from a declined transaction type (AAC) to an authorized one (ARQC), since it is reported that the algorithm for generating the AAC cryptogram is exactly the same as for the ARQC cryptogram. The Cryptogram Confusion attack can either be used in the phone lock-screen bypass attacks as in [135], or can be used for making transactions with locked cards, as in [146]. In the latter, if the cardholder enters the PIN in the chip-and-PIN mode incorrectly three times, it will restrict the card's functionality, by sending an invalid APDU value at the end of the EMV message. However, Yunusov was able to bypass this by changing the invalid response (6283: Selected File Invalidated) to a valid one (9000: Command Successfully Executed) indicating that it is capable of NFC communication, followed by CID modification.It is reported that this attack is effective on roughly 30% of cards. Contrarily, our experimental outcomes were unable to replicate this attack, signifying either that card issuers have fixed this issue or our specific card issuers are not vulnerable to this type of attack. Additional data logs detailing our unsuccessful replication attempt can be reviewed in Appendix A.4.

Two other attacks [16] [24] report the possibility of sending an unauthenticated cryptogram. The first attack [24] involves corrupting the AC on Visa cards when used with an offline reader in fDDA transactions. It is claimed that since the AC is not included in the SDAD in Visa, corrupted transactions are accepted by the offline reader. The second attack [16] discusses that the card does not authenticate the AC to the terminal in an offline contactless transaction with a Visa or an old Mastercard card, allowing criminals to trick the terminal into accepting an unauthentic offline transaction. None of these two attacks have been demonstrated.

3.4 Contactless Protocol Vulnerabilities

In this section, we will explore the weaknesses of the protocols based on the reported attacks for the main targeted protocols, specifically ISO 14443, the Entry Point, the Kernel 3 (Visa), and the Kernel 2 (Mastercard). The Entry Point protocol will be discussed in conjunction with the kernel protocols to

ensure a comprehensive and coherent analysis. It is crucial to highlight that this section does not discuss whether these vulnerabilities have been patched or resolved. The focus is instead on illustrating how specific messages can be actively manipulated by altering or flipping certain bits or passively intercepted for malicious purposes.

3.4.1 ISO14443 Vulnerabilities

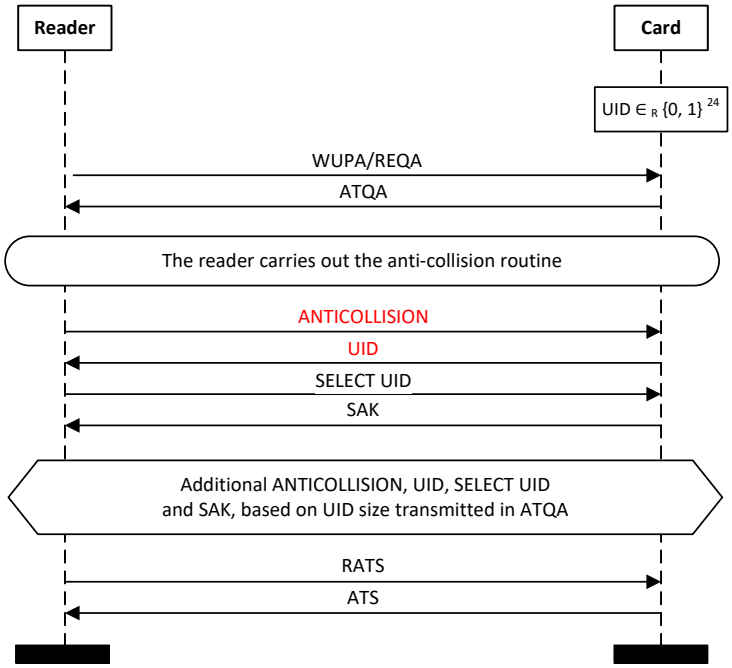


Figure 3.2: ISO14443 Vulnerabilities

As previously discussed in Section 2.2, the ISO14443 protocol is primarily designed for the identification of cards or objects in contactless payment systems. However, the ISO14443 has been the target of two attacks, visually depicted in Fig. 3.2 and expanded upon in Table 3.2.

In [96], it was discovered that the **anti-collision** process could be bypassed in the presence of another requesting entity (malicious application on the phone) within the field. Although the data is not directly modified, this vulnerability was attributed to a race condition that the malicious application wins. The other attack was explored in [109] where the **UID** within the ISO14443 system

Table 3.2: ISO14443 Vulnerabilities based on Attacks

Field	Vulnerability	Attack Goal	Modified	Demo
Anti-collision	Responds to a malicious requester	Data Leakage (Privacy Tracking) [96]	×	✓
UID	Can be modified by a rooted phone	Relay (Protocol Bypass)[109]	✓	✓

was found to be modifiable with the aid of a rooted phone to mimic a legitimate card which enabled attackers to bypass Visa relay protection.

3.4.2 Visa Vulnerabilities

As examined in Section 2.4, the protocol implementation of Visa is structured around Kernel 3 of EMV. Visa's system has been exposed to a variety of attacks, as illustrated in Fig. 3.3. A detailed description of these vulnerabilities is given in Table 3.3. Upon analyzing Fig. 3.3, it is evident that Visa has been a victim of attacks across all levels. The specific vulnerabilities of the messages are as follows:

Message (0): prior to initiating a Visa transaction, the "magic string" used in Apple Pay transit operations, can be employed on a standard terminal, that can convince the terminal that it is communicating with a transit operator terminal (with TTQ modifications). Instead of rejecting these bytes, the protocol perceives the reader as being in transit mode.

Message (5): the cardholder verification can be bypassed by altering the TTQ value, specifically to manipulate the CVM and ODA within the TTQ. The former indicates that no CVM is necessary, while the latter takes advantage of a transit operator-exclusive feature. Additionally, the Authorized Amount (AmountAuth) and Other Amount (AmountOther) can be modified. The currency is also reported to be alterable, as checks are predominantly conducted in the native currency. A fixed UN could also be sent through a compromised terminal.

Message (6): this message, which carries cryptogram information, might enable the sending of an invalid AC during fDDA transactions. In earlier versions of the specification, the AIP could be altered to downgrade from EMV mode to mag-stripe mode, however, support for this mode was eliminated by Visa in Kernel 3 specification Version 2.6 [42], thus it is not present in the latest specification. The CID is modifiable and checks on ATC values by issuers are limited. Moreover, the CTQ can be modified to set the CDCVM, fooling the

terminal into believing that the consumer device has performed CVM. Lastly, unencrypted Track 2 data can be requested and read by any unauthenticated terminal.

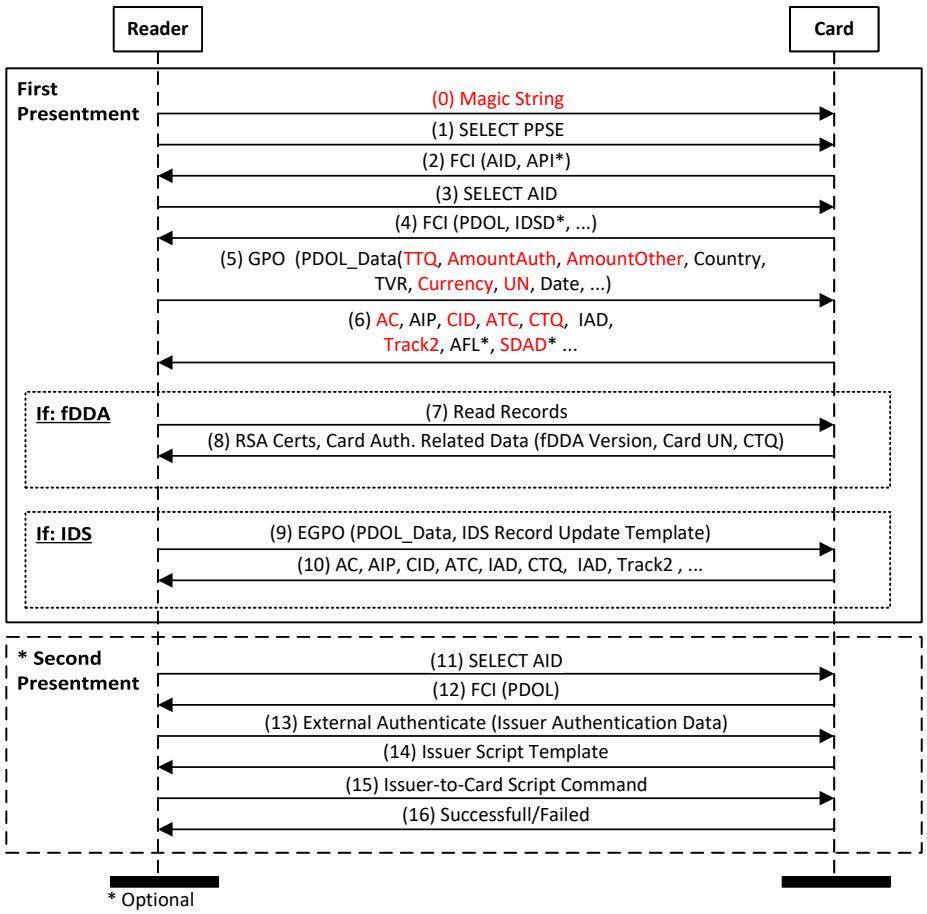


Figure 3.3: Visa Vulnerabilities

3.4.3 Mastercard Vulnerabilities

As previously introduced in Section 2.5, Mastercard's protocol employs Kernel 2 of EMV, which presents various vulnerabilities at different levels. A detailed overview of these vulnerabilities, including the targeted messages within the protocol subjected to documented attacks, is illustrated in Fig. 3.4, while Table 3.4 provides a more detailed breakdown. An analysis of Fig. 3.4 reveals that all stages of the Mastercard process, including the card, cardholder, and

Table 3.3: Visa Vulnerabilities based on Attacks

No.	Field	Vulnerability	Attack Goal	Modified	Demo
0	**Magic String**	Can be sent to mimic TFL transport	Lock-screen Bypass (GooglePay-Visa) [109, 144]	×	✓
5	**TTQ**	Can be modified (ODA for Online Authorization)	Lock-screen Bypass (GooglePay-Visa) [109, 144]	✓	✓
		Can be modified (CVM)	Lock-screen Bypass (GooglePay-Visa) [135]	✓	✓
			Limit Bypass [61]	✓	✓
	Amount Auth.	Can be modified via compromised PoS	Lock-screen Bypass (SamsungPay-Visa V1) [135, 144]	✓	✓
	Amount Other	Can be modified, Not being checked in SamsungPay zero-value req in transit	Lock-screen Bypass (SamsungPay-Visa V2) [109]	✓	×
	Currency	Not being checked in high-value transactions	Limit Bypass [32]	✓	×
	UN	Can send fixed UN with compromised terminal, No limitation in repeating	Pre-play [61]	✓	✓
6	**AC**	Send unauthenticated AC	Cryptogram Exploitation [24]	✓	×
			Cryptogram Exploitation [16]	✓	×
	AIP	Downgrade to Magstripe (old)	Card Replica [52]	✓	✓
	CID	Can be changed (from declined (AAC) to successful (ARQC))	Cryptogram Exploitation [146]	✓	✓
			Cryptogram Exploitation [135, 144]	✓	✓
	ATC	Not being checked by issuer	Pre-play [61]	✓	✓
			Cryptogram Exploitation [135, 144]	✓	✓
	CTQ	Can be modified (CDCVM)	Limit Bypass [16]	✓	✓
			Limit Bypass [61]	✓	✓
			(over the limit) Lock-screen Bypass [109]	✓	✓
	Track 2	Is sent in clear and can be requested via any unauthenticated NFC reader	Card Replica [59]	×	✓
			Card Replica [52]	×	✓
			Data Leakage [71]	×	✓
			Data Leakage [33]	×	✓
			Card Replica [102]	×	✓
	SDAD	Can be modified	Cryptogram Exploitation [16]	✓	×

transaction phases, are prone to attacks. The vulnerabilities of the messages are as follows:

Message (3): the AID of the card can be modified enabling the execution of attacks on a less secure AID with inherent vulnerabilities or allowing for the bypassing of the current secure AID.

Message (6): the AIP can be modified in such a way as to downgrade a more secure EMV transaction to a less secure mag-stripe transaction.

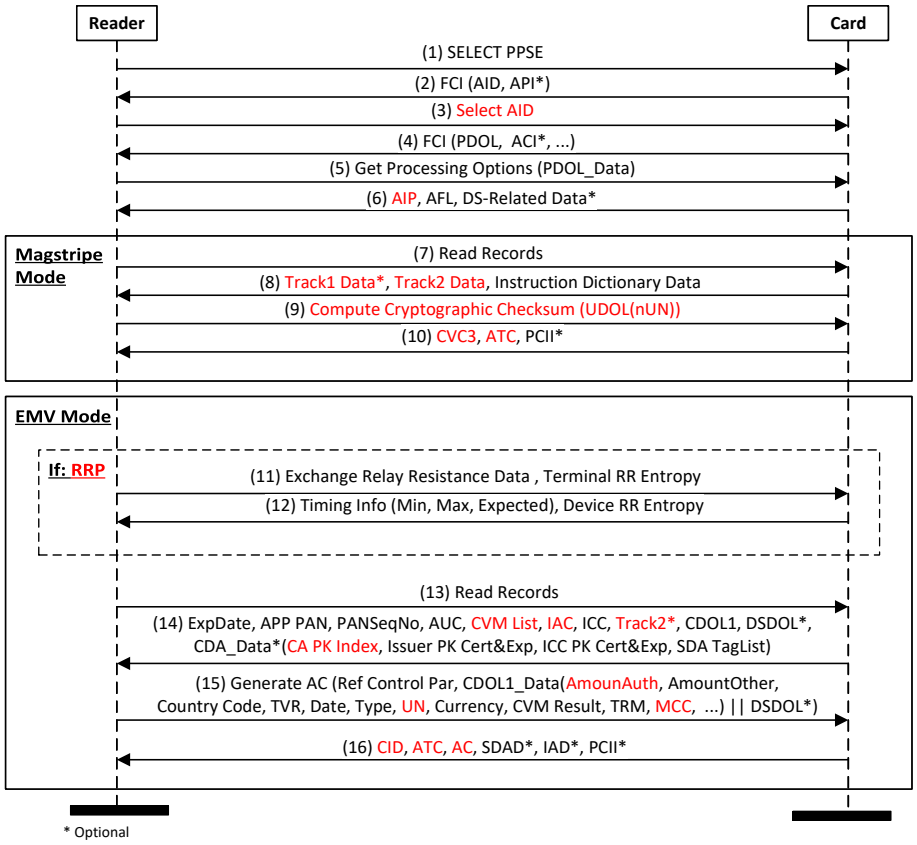


Figure 3.4: Mastercard Vulnerabilities

Message (8): Track 1 and 2 data are sent in a clear and unauthenticated way, that allows any unauthenticated terminal to request and read this data.

Message (9, 10): mag-stripe mode has several vulnerabilities. Due to the low entropy of the unpredictable number (nUN), usually 3 digits, CVC values can be pre-calculated. Furthermore, the Compute Cryptographic Checksum (CCC) message can be modified to bypass the high-value transaction unlocking requirements. ATC is also out of order and is not checked by many issuers.

Message (12): The RRP protocol timings were bypassed in EMV mode when payment card response times varied based on the card-reader distance, making the RRP protocol less effective for relay attacks.

Message (14): several values in this message, including IAC-Denial, CVM-List, and Certification Authority Public Key Index (CA PK Index) can be modified. Similar to mag-stripe mode, Track 2 data can be requested by

Table 3.4: Mastercard Vulnerabilities based on Attacks

No.	Field	Vulnerability	Attack Goal	Modify	Demo
3	**AID**	Can be modified	Limit Bypass [17]	✓	✓
			Limit Bypass [18]	✓	×
			Lock-screen Bypass (Samsung-MC) [144]	✓	✓
6	**AIP**	Can be modified	Pre-play [110]	✓	✓
			Lock-screen Bypass (GPay-MC) [135]	✓	✓
			Pre-play [52]	✓	✓
8	**Track1,2**	Is sent in clear and can be requested via any unauthenticated NFC reader	Card Replica [59]	×	✓
9	**CCC**	Can be modified to bypass "unlock" req.	Lock-screen Bypass (GPay-MC) [135]	✓	✓
	nUN	Has low entropy (3 digits)	Lock-screen Bypass (GPay-MC) [135]	×	✓
			Pre-play [110]	×	✓
10	**CVC3**	Can be pre-calculated (due to low entropy of nUN)	Lock-screen Bypass Bypass (GPay-MC) [135]	×	✓
			Pre-play [110]	×	✓
			Pre-play [52]	×	✓
	ATC	Not being checked by the issuer	Lock-screen Bypass (GPay-MC) [135, 145]	×	✓
			Pre-play [110]	×	✓
			Pre-play [52]	×	✓
12	**RRP**	Can be bypassed in specific positions	Relay [109]	×	✓
14	**CVMList**	Can be modified	Limit Bypass [18]	✓	✓
	IAC	Can be cleared (IAC-Denial)	Limit Bypass [18]	✓	✓
	Track2	Is sent in clear and can be requested via any unauthenticated NFC reader	Card Replica [59]	×	✓
			Data Leakage [71]	×	✓
			Data Leakage [33]	×	✓
			Card Replica [102]	×	✓
	CA PK Index	Can be modified to invalid index to bypass PKI checks	Limit Bypass [18]	✓	✓
15	**Amount Auth.**	Can be modified via compromised terminal	Lock-screen Bypass (Samsung-MC) [135]	✓	✓
			Lock-screen Bypass (Apple-MC) [135]	✓	✓
	UN	Can send fixed UN via compromised terminal, No limitation in repeating	Pre-play [61]	✓	✓
	MCC	can be modified via compromised terminal	Lock-screen Bypass (Samsung-MC) [135]	✓	✓
			Lock-screen Bypass (Apple-MC) [135]	✓	×
16	**CID**	Can be changed (from failed (AAC) to online (ARQC)	Lock-screen Bypass (Samsung-MC) [144]	✓	✓
	ATC	Not being checked by the issuer	Pre-play [61]	×	✓
	AC	Send unauthenticated AC	Cryptogram Exploitation [16]	×	×

unauthenticated readers.

Message (15): in this message, using a compromised terminal, the authorized amount (AmountAuth) can be modified. Furthermore, a fixed predictable UN can be sent, which shows there are no restrictions on sending similar UNs to the card. Also, the MCC value can be modified to mimic a transit operator terminal.

Message (16): similar to mag-stripe mode, ATC values are not checked by some issuers. Additionally, CID can be changed from a failed transaction to a successful one, which shows using of similar algorithms for both. There have also been suggestions that invalid ACs could be sent, although it has been reported only on old Mastercard cards and has not been demonstrated.

3.5 Observations

3.5.1 Visa versus Mastercard

Based on these vulnerabilities, here, we focus on two widely used payment networks; Visa and Mastercard, and compare the level of their vulnerability based on card-centric, cardholder-centric, and transaction-centric categorization, as discussed below. A comparison between the vulnerabilities of each protocol can be found in Table. 3.5.

Card-centric Attacks: When it comes to card-centric attacks, both Visa and Mastercard are susceptible. However, the vulnerabilities slightly vary. Although they were reported to be vulnerable to Anti-collision attacks, additionally, Visa's vulnerability lies within the ISO 14443 protocol, where the UID can be manipulated. In contrast, Mastercard's vulnerability comes from the EMV Entry Point protocol, where the AID can be modified.

Cardholder-centric Attacks: Interestingly, Visa appears to be more susceptible to cardholder-centric attacks than Mastercard. Numerous attacks target the modification of the TTQ and CTQ values, both containing cardholder verification data in Visa. Conversely, Mastercard only has a single demonstrated attack that aims to modify or clear the CVMList.

Transaction-centric Attacks: Mastercard experiences a higher frequency of transaction-related attacks. Although both Visa and Mastercard are vulnerable to the modification, interception, or exploitation of transaction data (like Track1/2, UN, ATC, AmountAuth, CID, and AC), Mastercard faces additional threats. In Visa, the Currency, AmountOther, and SDAD values are at risk,

whereas in Mastercard, the MCC, IAC, AIP, CA PK Index, and all the mag-stripe related data including nUN, CVC, CCC could be exploited for malicious purposes.

It is crucial to recognize that these comparisons do not inherently imply that one protocol is more secure than the other. To conduct a more precise assessment, it is essential to consider various factors, some of which are outlined below. Firstly, the higher prevalence of Visa compared to Mastercard, as reported by Statista [127] based on global card brands' transactions from 2014 to 2022, may lead to an increased number of reported attacks within a specific layer since Visa has shown greater prevalence in the market. This can also be due to the variances in the behaviour of Visa and Mastercard across different countries, which could contribute to distinct disparities in each network. Secondly, the visibility of a particular field in a transaction, its presence in the Application Cryptogram (AC), Integrated Circuit Card (ICC) certificates, and/or Static Data Authentication Data (SDAD) can impact the feasibility of modifying a specific field and the likelihood of vulnerabilities being detected. For instance, the CTQ field plays a crucial role in attacks on Visa. As discussed in this chapter, it can be modified and is not directly authenticated in the AC. In contrast, in Mastercard, the Cardholder Verification Method (CVM) lists, which would be edited in an attack instead of the CTQ, are present in the ICC certificate, and even a PoS system can detect an attempted change in them (unless making the terminal to ignore the checks as shown in Basin et. al. work [18]). Different business models may also be evident; it appears that Mastercard systematically verifies cryptographically-ensured fields (e.g., the AC) or other proprietary fields (e.g., the Issuer Application Data or IAD) [18], while Visa may not consistently verify these fields [109]. Moreover, the severity and feasibility of each attack in real-world scenarios, along with their frequency of occurrence, play significant roles in evaluating their respective security measures. Indeed, what appears to be a minor vulnerability could have a significant impact if the exploit is easily repeatable and frequently occurs. Finally, Visa and Mastercard have different functionalities that should be considered (e.g., the vulnerable mag-stripe mode in Mastercard). These considerations underscore the necessity for a comprehensive evaluation when analyzing security issues within these payment protocols.

Table 3.5: Comparative Analysis of Vulnerabilities: Visa versus Mastercard

	Visa	**Mastercard**
Card-centric	ISO14443: UID, Anti-collision	EMV Entry Point: AID ISO14443: Anti-collision
Cardholder-centric	TTQ (CVM), CTQ (CDCVM), Offline PIN (patched)	CVMLits
Transaction-centric	Track2, UN, ATC, AmountAuth, CID, AC, Currency, AmountOther, SDAD, TTQ (ODA)	Track1, Track2, UN, ATC, AmountAuth, CID, AC, MCC, IAC, AIP, CA PK Index, nUN, CVC, CCC

3.5.2 Failures

Considering these vulnerabilities, here, we categorize the ways contactless payment systems have failed based on the literature as follows:

Offline Mode: While no successful attacks on offline transactions have been demonstrated, there remains a potential vulnerability. Specifically, an attacker could send unauthenticated cryptograms, especially during Visa fDDA transactions. While these transactions are likely to be rejected upon subsequent online verification, this delay can provide the attacker enough time to run away with the goods.

Mag-stripe Mode: Vulnerabilities in this category are due to the potential downgrade or manipulation of payment devices to function in the less secure mag-stripe mode. Although Visa has discontinued support for the mag-stripe mode as previously mentioned [42], Mastercard persists in allowing this transaction method in its latest specification. This leaves room for exploitation, especially considering that the entropy of the unpredictable number in this mode for Mastercard is low.

Unencrypted Data: Both Visa and Mastercard transmit critical data, such as account numbers and card expiry dates (found in Track 1 and Track 2 data), without encryption. This exposes the data to interception by any unauthenticated terminal, potentially leading to data breaches.

Unauthenticated Data: A significant portion of vulnerabilities in Visa and Mastercard protocols stem from the fact that certain card-generated data isn't authenticated online by card issuers. This oversight may enable data manipulation or exploitation for malicious purposes. Notable instances include ATC, CTQ, and CID for Visa, and ATC, CA PK Index, IAC, AIP, and CID for Mastercard. For instance, transactions should be declined by issuers when the CDCVM is set by a card since this CVM is intended only for NFC-enabled

devices, not cards. Another point of concern is the inconsistency between the CVM list and the CVM result.

Unauthenticated/Compromised Terminal: The current systems lack robust terminal authentication. This deficiency is evident in modes like ApplePay's transit mode, where simply sending a specific string can simulate a transport terminal due to inadequate terminal authentication measures. Furthermore, both Visa and Mastercard are vulnerable to high-impact attacks when terminals get compromised. In such scenarios, critical security parameters like AmountAuth, UN, and MCC can be altered, due to the absence of integrity checks.

Ineffective Relay Protection: Although Visa and Mastercard have incorporated relay protection protocols to prevent relay attacks, these safeguards have been circumvented, highlighting their inadequacy. This emphasizes the persistent risk associated with relay attacks.

3.5.3 Countermeasures

Addressing the vulnerability categories specified in Section 3.5.2 can be interpreted as introducing countermeasures. Advocating for the phasing out of the less secure mag-stripe mode in Mastercard[6], mimicking Visa's proactive step, or at the very least incorporating an unpredictable number with higher entropy, is proposed. As for the issue of unencrypted data, mandatory encryption of sensitive data, such as account numbers and expiration dates, is recommended. Further, implementing online authentication mechanisms for card-generated data by card issuers can mitigate risks associated with unauthenticated data. For unauthenticated or compromised terminals, developing robust terminal authentication processes as well as establishing integrity checks are required. Lastly, to counter ineffective relay protection, a reconsideration of the design of current relay protection protocols is advised to strengthen defences against relay attacks. A viable solution is the OPay system, elaborated upon in Chapter 5.

However, mitigating these vulnerabilities can be complicated and challenging, given the potential unwillingness of the involved parties to take responsibility. For instance, consider the lock-screen bypass attack on Apple Pay and Visa, as in [109]. Following the disclosure of the researchers' findings, both entities delivered conflicting responses. Apple suggested that Visa should amplify

[6]Mastercard's plan is to phase out mag-stripe cards in Europe in 2024, and plans to remove requirements for U.S. banks to issue chip cards with mag-stripe from 2027 [91].

its fraud detection mechanism and add more verification steps, whereas Visa claimed that the problem was unique to Apple, suggesting a need for corrective measures in Apple Pay. Regrettably, neither entity took any steps, and as of August 2023, the attack remains active.

Considering the complex nature of the different cards and systems and the potential lack of cooperation among the stakeholders, a pressing demand for strategies to navigate these challenges emerges. This calls for a standardized testing process to analyze both the viability of the attacks and the prospective countermeasures that include all involved parties.

It's important to note that the newly introduced single contactless kernel, Kernel 8 (described in Section 2.6), can alleviate some of these vulnerabilities. For instance, the Privacy Protection feature aims to protect the privacy of the card or payment application's identity during transactions. Nonetheless, further exploration of practical countermeasures remains vital to alleviate the identified vulnerabilities comprehensively.

3.6 Conclusion

Through categorization and analysis, this chapter presents a thorough exploration of attacks in contactless payment systems. The seven-fold classification of attacks targeting three distinct levels provides a valuable systematization for understanding the vulnerabilities in major protocols such as ISO14443, EMV Entry Point, Visa, and Mastercard. Our comparative study of Visa and Mastercard highlights distinct vulnerabilities across cardholder-centric and transaction-centric levels. Further examination into the causes behind these vulnerabilities underscores the failure of these protocols to sufficiently secure the user's data and transactions. In response to these identified flaws, by considering the potential challenges, the chapter concluded with a set of proposed countermeasures, paving the way toward a more secure contactless payment system.

Chapter 4

Security Analysis of Mobile Point-of-Sale Terminals

4.1 Overview

The increasing prevalence of Card Present (CP) transactions has driven the growth of mobile Point-of-Sale (mPoS) terminals. These compact, wireless, and low-cost terminals allow merchants to process transactions conveniently by utilizing a mobile phone. In this chapter, we analyze the security implications of mPoS terminals with a focus on studying merchants' mobile phones as a key component in the mPoS ecosystem. Our examination covers the security aspects of the mobile phone's communication with the mPoS terminal and the payment provider server, as well as the security risks in the mobile phone application itself. We perform an eavesdropping attack to reveal the cryptographic keys in the BLE (Bluetooth Low Energy) communication between the mPoS terminal and the merchant phone, execute a man-in-the-middle (MITM) attack to tamper with the mPoS terminal messages transmitted between the mPoS terminal and the payment provider server, and reverse engineer the mobile phone application to disable the security features that are controlled by the mobile phone.

4.2 Introduction

As stated in Section 1.2.1, traditionally, PoS terminals have been used to process CP transactions. These terminals are typically large, fixed devices that are found in retail stores and other locations where goods and services are sold. They are connected to a payment processor through a wired or wireless network.

However, with the growing demand for more flexible and cost-effective payment solutions, mobile PoS (mPoS) terminals have emerged as an alternative to traditional PoS terminals due to their flexibility and affordability, especially for small businesses. Examples are Sumup [133], Square [124], and iZettle [78]. These terminals are small, compact, low-cost, wireless, and easy to configure, requiring a few simple steps. They are equipped to accept various payment methods such as debit/credit/prepaid cards with magnetic strips (mag-stripe) or embedded chips, contactless payments through mobile wallets, QR codes, and/or cash and checks [56]. They offer the ability for anyone with a bank account to establish their own payment terminal, mostly without requiring a business account or a fixed contract.

Although they provide convenience for merchants and customers, they raise potential risks that can be exploited for malicious purposes. This can include holding an mPoS terminal near a victim's payment device (credit/debit card or NFC-enabled devices such as smartphones or wearable devices (e.g., smartwatches) without their knowledge, in conjunction with other emulation hardware, to perform malicious attacks, as explained in details in chapter 3.

The management of these terminals is usually done with a mobile device, such as a mobile phone or tablet, which plays a crucial role in various aspects of the transaction process, including the establishment of a Bluetooth connection with the mPoS terminal, the connection to the payment provider server over the internet, and the installation of an application on the device to manage the mPoS terminal. In this chapter, the potential security risks and vulnerabilities of mPoS terminals are analyzed with a focus on the involvement of mobile phones in their management, which is owned by the merchant. Specifically, the security aspects of the communication between the mobile phone and the mPoS terminal, the communication between the mobile phone and the payment provider server, and the mobile phone application itself are examined. The security of the Bluetooth Low Energy (BLE) communication between the mobile phone and the mPoS terminal is analyzed, and methods for revealing the cryptographic keys used in this communication are explored. Furthermore, a MITM attack is performed to demonstrate the vulnerability of the communication between the mobile phone and the payment provider server. Additionally, the feasibility of reverse engineering the mobile phone application code is shown, and the modification of the security features of the mPoS terminals controlled by the mobile phone is demonstrated. We summarize our contributions as follows:

- Performing an eavesdropping attack on the BLE communication between the mobile phone and the mPoS terminal to extract the cryptographic keys used for communication;
- Performing a MITM attack between the mobile phone and the payment server to intercept and tamper with the messages to be displayed on the terminal;
- Demonstrating the feasibility of reverse engineering the mobile phone application code and the alteration of the security features of the mPoS terminals that are controlled by the mobile phone.

This chapter employs the terms *card reader*, *terminal*, and *mPoS terminal* interchangeably. Moreover, these vulnerabilities are not inherently tied to the EMV protocol itself but rather to external protocols within the system, namely Bluetooth and HTTPS. For example, if the Bluetooth connection between the PoS and the mobile phone is compromised, it becomes evident that the displayed information on the PoS can be manipulated independently from what is shown on the phone.

The rest of the chapter is organized as follows. In Section 4.3, we provide the background and the related work on studying the mPoS terminals vulnerabilities. Section 4.4 explains encryption security, with a focus on the BLE communication between the mPoS terminal and the mobile phone. Section 4.5 explains network security, with a focus on the security vulnerabilities of the HTTP communication between the mobile phone and the payment server. In Section 4.6, we investigate the mobile application installed on the mobile phone and demonstrate the feasibility of bypassing the security features, followed by a discussion in Section 4.7. Finally, we conclude the chapter in Section 4.8.

4.3 Background and Related Work

The installation of an mPoS terminal requires a series of straightforward steps. These steps include purchasing the device, which can vary in price based on its features (with options starting as low as £19), registering for an online account (usually done via the vendor website), installing the corresponding application on the merchant's mobile phone, pairing the phone with the terminal, and finally, making transactions.

The ecosystem of mPoS terminals and their communication with various entities in transactions are depicted in Fig. 4.1, which is a subset of the payment

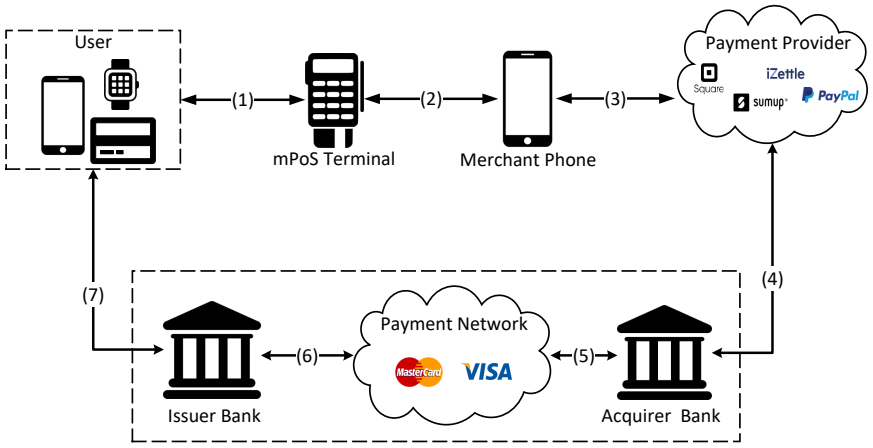


Figure 4.1: Mobile Point-of-Sale (mPoS) Terminals Ecosystem

ecosystem shown in Fig. 1.1. The mPoS terminal is operated by a mobile phone, owned by the merchant. The merchant downloads an application on their mobile phone and uses it to connect to the mPoS terminal. This enables the merchant to initiate and request payments. When the payment is sent from the merchant's mobile phone to the mPoS terminal, the user is ready to pay.

As shown in Fig. 4.1, the user has the option to make a payment transaction through either a contactless or chip-and-PIN method by tapping, inserting, or swiping their payment device against the mPoS terminal (1). The payment is then transmitted from the mPoS terminal to the merchant's mobile phone through Bluetooth communication (2). The transaction information is then transmitted from the merchant's mobile phone to the payment provider server for authorization (3). The payment provider, in turn, communicates with the acquirer bank to verify the transaction details and ensure its security and accuracy (4). The acquirer verifies the authenticity of the customer's payment card and checks the available funds with the payment network (5), which communicates with the card issuer (6). Upon receiving approval from the card issuer, the customer's account is charged, and the customer is notified (7). The merchant's account is credited, and the notification is propagated all the way back to the merchant's mobile phone.

The mPoS terminals have been the subject of security studies in the past decade. One of the first studies, by Frisby et. al. [58] in 2012, investigated the smartphone-based PoS systems that consist of a software application combined with an audio-jack mag-stripe reader (AMSR) on a smartphone. The study focused on mPoS systems that relied on a smartphone, incorporating an AMSR

and a corresponding application running on an Android smartphone. The security assessment concluded that any application running on the smartphone could potentially disable the mag-stripe reader and obtain confidential cryptographic keys. However, the architecture of mPoS terminals has since evolved, and the current study is not centred around AMSR but shifts the focus from audio-jack mag-stripe smartphone-based PoS systems to mPoS terminals that are controlled via smartphones.

A subsequent study on mPoS terminals is by Mellen et. al. [97] where they demonstrated potential attack vectors for Square [124] mPoS terminals, both in the software and hardware. In software, their research found security weaknesses in the old Square terminals, which were later deprecated, and discovered vulnerabilities in the encrypted Square reader S4 model and Square registration application, which have since been addressed. In the hardware, the researchers discovered that the Square reader devices used a chip for point-of-swipe encryption, but were able to bypass the encryption by jumping the connection from the magnetic head reader to the headphone jack input or by crushing the encryption chip. The attack tool, called Swordphish, was developed to record unencrypted swipes and transmit the credit card information to an external server.

In another study published in [84], the security of mPoS terminals, with a specific emphasis on the Miura [134] Shuttle chip-and-PIN reader, was thoroughly investigated. The researchers demonstrated the capability of performing arbitrary code execution as a root user on the device, utilizing both the USB and Bluetooth interfaces. Additionally, they exhibited how they could gain root access to the terminal via the chip-and-PIN mode, thereby manipulating the display and keyboard of the device to elicit the entry of the user's PIN, by changing the displayed message to "Try Again" and downgrading to magnetic stripe (mag-stripe) mode. However, this vulnerability was remediated by 2014.

In 2018, researchers in [60] conducted a follow-up investigation, exploiting a vulnerability that existed at that time through the Bluetooth interface. It was found that the SumUp [133] terminal transmitted commands in plaintext over Bluetooth, thereby allowing for the sending of arbitrary commands and tampering of amounts, following the reverse engineering of the terminal's characteristics and functions. As a result, researchers were able to perform a similar attack vector, as outlined in [84], by manipulating the displayed messages to prompt the user to swipe their card with a message that reads "Please Swipe Card". Our subsequent analysis of transaction data collected from

SumUp terminals, however, revealed that the vulnerability had been addressed by the vendor with the implementation of encryption for all messages. More details will be provided in Section 4.4. Thus, the demonstrated attack vector is no longer viable, as a successful attacker would require knowledge of the encryption key to send valid messages to the card reader through Bluetooth communication. The researchers also explored the manipulation of amounts in magstripe mode transactions through the forcing of card swiping. Finally, the study highlights the use of a tamper detection circuit in the tested terminals, which would render the device inoperable in the event of attempted tampering.

Having previously addressed vulnerabilities from various angles on different mPoS terminals, in this chapter, we explore the mPoS terminal ecosystem from a novel standpoint, examining the capacity of merchant's mobile phones to initiate attacks as it is a crucial part of the mPoS ecosystem. This study involves a comprehensive analysis of the mobile application and the communication protocols between the mPoS terminal, merchant phone, and payment provider server. The aim of the analysis is to identify and examine security weaknesses at various layers, in order to provide insights into the mitigation of associated risks.

4.4 Encryption Security

The deployment of an mPoS terminal requires the establishment of a wireless communication channel with the merchant's device, typically a mobile phone which is owned by the merchant. Bluetooth Low Energy (BLE) is a widely used technology for this purpose. The merchant first pairs an mPoS terminal with their mobile phone and uses that established communication link to send and receive transactions to/from the mPoS terminal. However, it is critical to consider the security implications of this communication channel, as exploitation of vulnerabilities can result in extracting the cryptographic keys. As previously stated, the attack vector described in [60] is no longer viable; our analysis of Bluetooth traffic contradicts the findings in [60], where certain commands sent to the SumUp terminal were discovered in plaintext. Subsequent security improvements made to the SumUp platform have made both packet analysis and arbitrary command execution more challenging since all the packets on the BLE communication are encrypted now. An example of the difference in the encryption between the Write Command values in [60] and our recent data collection is shown in Fig. 4.2.

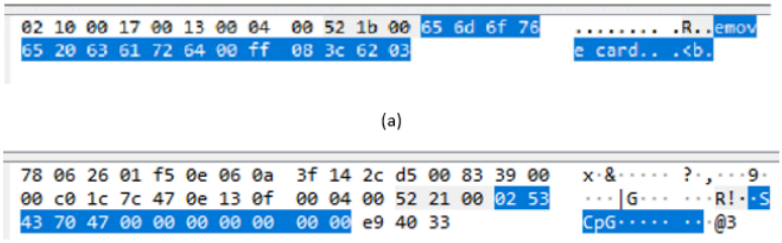


Figure 4.2: (a) Write command example found in [60] (b) Write command example captured in this chapter

To carry out the arbitrary command execution attack, an attacker would need knowledge of the encryption key in order to send valid messages to the mPoS terminal through Bluetooth communication. In this section, we first provide background information on BLE communication with a focus on the pairing session and then demonstrate how it is possible to capture the cryptographic keys of the BLE communication by exploiting existing vulnerabilities in the pairing session between the mPoS terminal and the merchant's mobile phone.

4.4.1 BLE Communication

The BLE protocol stack is comprised of three main architectural layers: the Controller, Host, and Application. The Host Controller Interface (HCI) serves as a bridge between the Host and Controller. The Security Manager Protocol (SMP) located in the Host layer is of particular importance in this context, as it is responsible for establishing secure connections and facilitating secure data exchange between devices. SMP outlines the procedures for pairing, authentication, and encryption of links between devices. During the pairing process, keys are generated for encrypting links and shared through a key distribution protocol for future connections and verification of data. The two devices involved in pairing are differentiated as the initiating device and the responding device. Here, the initiating device is the merchant's mobile phone and the responding device is the mPoS terminal.

The BLE Pairing procedure is shown in Fig. 4.3. Based on the BLE specification [120], the SMP carries out pairing in three phases: phase 1, phase 2, and phase 3. In phase 1, the devices engage in a Pairing Feature Exchange using the SMP Pairing Request and Pairing Response commands. During this

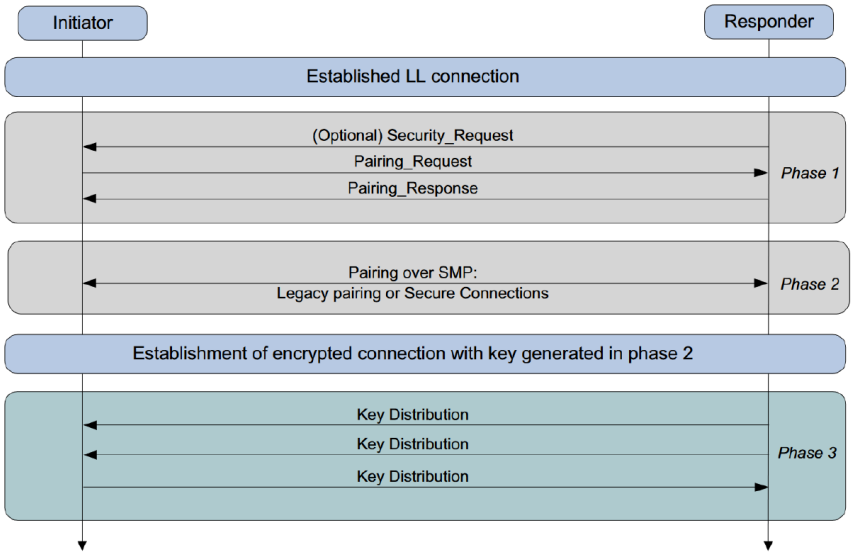


Figure 4.3: BLE Pairing Phases [120]

exchange, information such as Input/Output (I/O) capability, Out-of-Band (OOB) data flags, Bonding flags, MITM protection, and Secure Connection (SC) requirements are shared between the devices. The keypress (KP) flag is only relevant in the Passkey Entry protocol and is ignored in other protocols. Based on this information, both devices determine their I/O capabilities and select the appropriate pairing mechanism for use in the next phase of the pairing process, according to the mapping table specified in the BLE specification.

In phase 2 of the pairing process, the devices utilize the information exchanged in the Pairing Feature Exchange to determine the suitable pairing mechanism, either Low Energy Legacy (LE Legacy) pairing or Secure Connection (SC) pairing.

In **LE Legacy** pairing, the devices exchange a Temporary Key (TK) and use it to create a Short Term Key (STK) which is used to encrypt the connection. If the I/O capabilities of a device, either the initiating or responding device, has a display capability, then it will display a randomly generated passkey value between “000000” and “999999”. The other device should have an input capability like a keyboard so a user can input the value displayed for the TK. If the I/O capabilities of both the initiating and responding devices do not have display capabilities but only have a keyboard, the user needs to guarantee that the TKs between the initiating and responding devices are the same.

This is a special case for Passkey Entry. After the generation of the TK, it is then combined with two random numbers to produce the STK; *Mrand* for the initiating device, *Srand* for the responding device. The *Mconfirm* and *Sconfirm* are 128-bit confirmation values that can be calculated using the confirm value generation function *c1*. The detail for this function is out of the scope of this research and can be found in Bluetooth Specification [120]. The security of this process depends greatly on the pairing method used to exchange the TK. In Legacy Pairing, the pairing method can be Just Works, Out of Band (OOB), or Passkey. In Just Works, the TK is set to zero. In OOB, the TK is exchanged using a different wireless technology such as NFC. In Passkey, the TK is a 6-digit number that is passed between the devices by the user.

In LE **Secure Connection**, instead of using a TK and STK, LE Secure Connections use a single Long Term Key (LTK) to encrypt the connection. This LTK is generated and exchanged using the Elliptic Curve Diffie Hellman (ECDH) protocol. In addition to supporting the pairing methods in the LE Legacy, it also supports the Numeric Comparison pairing method. It is similar to Just Works but adds another step at the end. Once the devices confirm that the confirmation values match, then both devices will independently generate a final 6-digit confirmation value using nonces. They both then display their calculated values to the user and the user manually checks both values match and confirms the connection.

In phase 3, the devices use the secure communication channel established in the previous phase to share the LTKs which will be used for link encryption. Each LTK is a 128-bit random number that may be generated along with a 16-bit Encrypted Diversifier (EDIV) and 64-bit Random Number (Rand) by both the slave and master device. The exact function of EDIV and Rand keys may vary depending on the implementation of the BLE protocol, but they are typically used to identify or derive the LTK for future connections. In order to conserve energy and storage, the slave device may not retain these values, leaving the responsibility of encrypting future communications to the master device, which in this case is the smartphone.

4.4.2 Eavesdropping to Extract Cryptographic Keys

The attacker, who may be a malicious merchant or an eavesdropper, can extract the cryptographic keys by capturing the pairing session between the mPoS terminal and the merchant's mobile phone. These keys are then used to carry out

various attacks. Malicious merchants can capture their phone's pairing session with their terminal during the initial BLE communication setup to obtain the cryptographic keys. These keys can then be utilized to access future transaction data exchanged between the phone and the terminal. An eavesdropper can also sniff the established BLE communication to compromise the encryption. As demonstrated in [112], the attacker can exploit the vulnerability of the BLE communication by jamming the connection, which forces the master and slave to reconnect and establish a new pairing session. During this process, the eavesdropper can inject appropriate control packets to initiate a key renegotiation to obtain the keys. Our proposed model takes advantage of the vulnerability present in the BLE communication between the merchant's phone and the mPoS terminal without requiring physical access to the mPoS terminal.

Eavesdropping: There are two primary methods for eavesdropping on BLE traffic: using the HCI Snoop Log on the merchant's mobile phone and using over-the-air Bluetooth sniffers. The HCI Snoop Log approach involves capturing and analyzing the HCI data packets on the merchant's Android phone, which can provide detailed information about the BLE communication between the phone and other devices. The over-the-air Bluetooth sniffers, on the other hand, capture BLE communication in the air by using specialized hardware and software. This approach is useful for monitoring and analyzing the Bluetooth traffic between multiple devices over a larger area. Both of these approaches have their own advantages and disadvantages and it depends on the specific requirements of the task and the environment in which it is being performed.

The utilization of HCI snoop logs, which requires the *Developers Options* setting to be enabled on the Android phone, offers several advantages. Firstly, the HCI snoop log is immune to missing packets during the capture process, which is a prevalent issue with over-the-air Bluetooth sniffers. Secondly, as the HCI protocol is situated above the Link Layer (LL) in the Bluetooth protocol stack, the contents of all packets are already decrypted by the LL. This results in a more straightforward analysis of the packets, as they are not impacted by the encryption performed by the LL. However, it has a limitation for some of the mPoS terminals, such as Square [124], that is equipped with the ability to recognize whether Developer Options are enabled on the smartphone, thereby disabling any transactions during this period. As a result, over-the-air Bluetooth sniffers would be a better choice for these mPoS terminals. We used the combination of HCI Snoop Log and Bluefruit BLE sniffer [1] to eavesdrop

on the pairing session of the mPoS terminal's BLE communication with an Android phone.

We used Pixel6 as our phone and tested SumUp Air and Square mPoS terminals to capture their pairing session with the phone. The pairing session of the Square [124] terminal is very similar to the SumUp [133] terminal. Hence, for our proof-of-concept, we show the pairing session for a SumUp terminal in Fig. 4.4, with detailed Pairing Request and Pairing Response shown in Table 4.1.

Extracting Cryptographic Keys: The pairing request, as depicted in Fig. 4.4, is initiated by the smartphone and details the desired parameters for the BLE connection. This includes the type of pairing, the I/O capabilities of both devices (the keyboard and display), the request for bonding for future connections, and the demand for a secure connection with MITM protection. The Max Encryption Size field of the request is set to 16, and the Initiator Key Distribution and Responder Key Distribution fields specify that all of the encryption keys (LTK, Identity Key (IRK), Signature Key (CSRK), and Link Key) should be distributed to both devices. This ensures that both the smartphone and the mPoS terminal have all of the necessary keys for secure and encrypted communication.

However, the response from the SumUp card reader to the pairing request is surprising in that it indicates a lack of I/O capabilities despite having both a keyboard and a display. Additionally, the respondent refuses to establish a secure connection and protection against MITM attacks. As a result, **LE Legacy** pairing will be used. The Initiator Key Distribution and Responder Key Distribution fields in the response specify that only the Encryption Key (LTK) and Id Key (IRK) will be shared between the devices, whereas the Signature Key (CSRK) and Link Key will not be exchanged.

It is determined from the mapping of I/O capabilities to the key generation method in the BLE specification (as specified in Table 2.8 of the Bluetooth Core Specification v5.3 [120]) that, given the initiator has a keyboard and display and the responder claims to have no input or output capabilities, the ***Just Works-Unauthenticated*** key generation method will be employed. The utilization of the Just Works pairing method results in the generation of the TK and STK. The Just Works STK generation method provides no protection against eavesdropping or MITM attacks during the pairing process. Both devices set the TK value utilized in the authentication mechanism to **zero**, leading to a lack of protection against such attacks. The STK is not explicitly

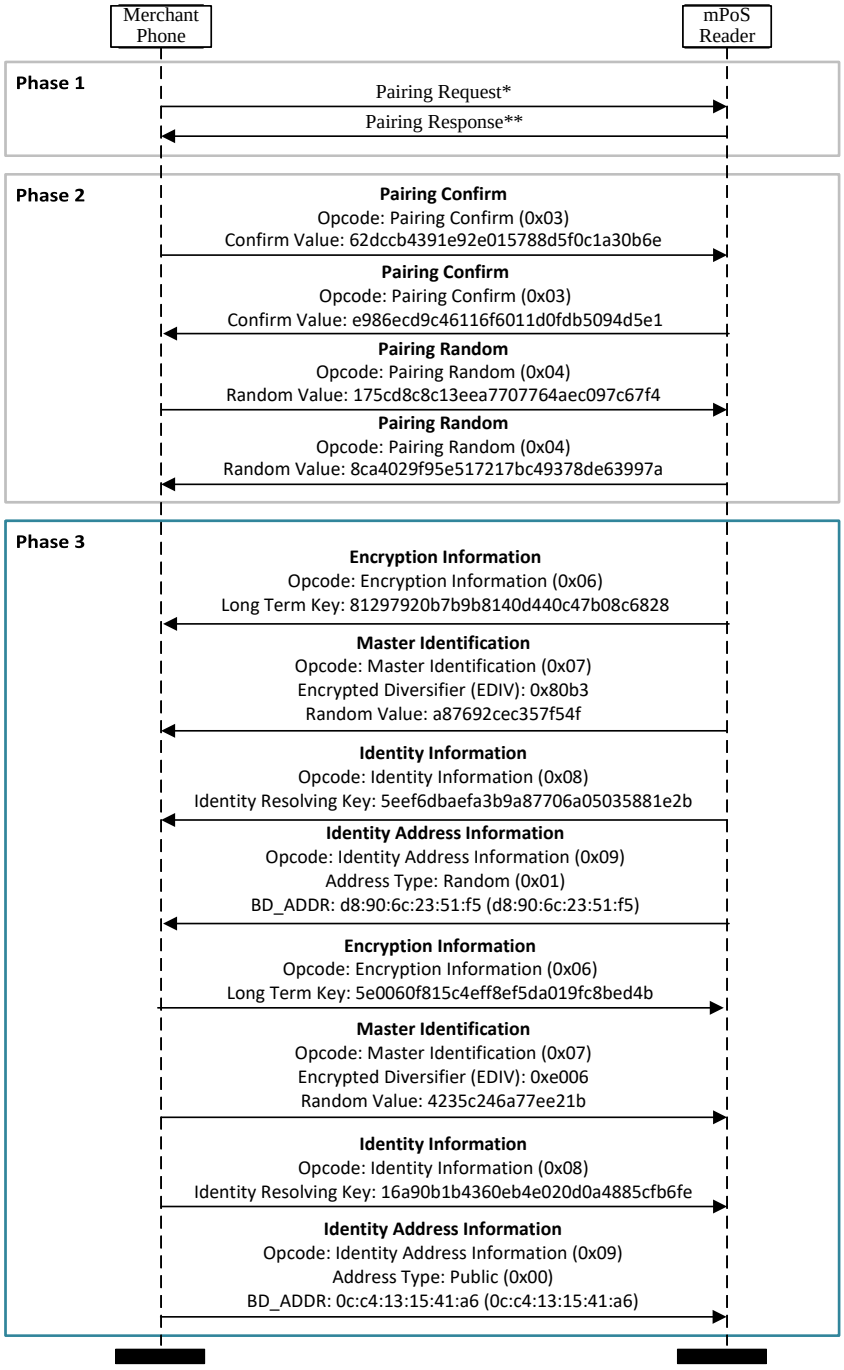


Figure 4.4: Pairing Session: SumUp Card Reader

Table 4.1: Pairing Request and Response: SumUp Card Reader

Field	**Pairing Request Value**	**Pairing Request Meaning**	**Pairing Response Value**	**Pairing Response Meaning**
Code	0x01	Pairing Request	0x02	Pairing Response
I/O	0x04	Keyboard/Display	0x03	No I/O
OOB	0x00	NOT Present	0x00	NOT Present
Authentication Request				
Bonding	0x1	Bonding	0x1	Bonding
MITM	1	True	0	False
SC	1	True	0	False
KP	0	False	0	False
Reserved	0x0	-	0x0	-
Max Enc.	16	Max Enc. Size	16	Max Enc. Size
Initiator Key Distribution				
LTK	1	True	1	True
IRK	1	True	1	True
CSRK	1	True	0	False
Link Key	1	True	0	False
Reserved	0x0	-	0x0	-
Responder Key Distribution				
LTK	1	True	1	True
IRK	1	True	1	True
CSRK	1	True	0	False
Link Key	1	True	0	False
Reserved	0x0	-	0x0	-

shared between the devices, rather the participating devices share random values and calculate the STK individually.

Due to the lack of utilization of the mPoS terminal's keyboard and display for a secure pairing method, the attacker can have access to the distributed keys in phase 3, as shown in Fig. 4.4. The access to security keys used in a LE Legacy pairing session by an attacker grants them the ability to eavesdrop on the data being transmitted between the two devices. This is because these keys are used to encrypt and secure communication, and having access to them would enable the attacker to decrypt the data and have access to it. For instance, if the attacker possesses the LTK, they could use it to encrypt the data exchanged between the two devices, allowing them to intercept and manipulate the data. Crackle [111] is one of the tools that can be used for this purpose. With the "Decrypt with LTK" feature, crackle uses a user-supplied LTK to decrypt communications between a master and slave.

Not utilizing the I/O capabilities for secure pairing is not common practice across all mPoS terminals. The examination of the SumUp Air mPoS terminal

in this study revealed that it does not employ such mechanisms, in contrast to other terminals like iZettle, which incorporate secure pairing techniques. Specifically, iZettle's method involves the presentation of a numerical value on the terminal's display, which the user must then confirm as matching the corresponding value on their paired device [77].

4.5 Network Security

The implementation of a mobile application on a smartphone connected to an mPoS terminal requires interaction with servers of the payment service providers through the Internet. In this section, we investigate the analysis of decrypted Hypertext Transfer Protocol Secure (HTTPS) packets and the feasibility of modifying these packets. The subsequent sections present the specifics of our intercepted network traffic, followed by a demonstration of a tampering attack on this traffic, serving as proof of concept for MITM attacks.

4.5.1 HTTPS Interception

The merchant's mobile phone uses HTTPS packets to communicate with payment providers over the Internet. This protocol employs Transport Layer Security (TLS) to encrypt network traffic. In order to gain access to the contents of these packets, a MITM attack is employed using a proxy server. The proxy server is able to intercept and decrypt the HTTPS packets, as the smartphone establishes a secure connection with it, believing it to be the intended recipient of the network traffic. The proxy server subsequently forwards the packets to the payment server. Details of communication over the course of a transaction for a SumUp terminal can be seen in Fig. 4.5. As shown in this figure, a transaction begins with a Checkout Request from the merchant's mobile phone, which requests the appropriate resources to display in the application during the transaction from the payment server. Other information in this request includes the currency, transaction amount, location and mPoS terminal device information, which is sent to the SumUp device for logging and handling purposes. For example, the transaction will fail and the sequence will end if the battery level of the terminal is too low. Continuing from the Checkout Request is a Transaction Request, where the beginning of the transaction is requested from a payment endpoint within SumUp's payment server. This is also the point at which the merchant's mobile phone begins to act as a proxy

for communications between the terminal and payment server, which exchange messages without the SumUp application's influence. After this response to the transaction request, we then see four or five request-response pairs to and from the payment endpoint, depending on the payment method (chip-and-PIN or contactless). After successful payment, the transaction ends with a response from the payment endpoint and a value *stop*. The SumUp application processes this action to end the transaction and reject any other responses from the terminal. The transaction officially ends when the merchant phone sends two messages to the terminal on behalf of the payment server, signalling a successful closure of the transaction.

In our attack scenario, the Mitmproxy tool [99] is utilized as the proxy server on a desktop computer to perform a MITM attack between the SumUp application and the payment server. This tool is designed as an interactive, SSL/TLS-capable intercepting proxy for HTTP/1, HTTP/2, and web sockets, as it allows the attacker to monitor, capture and alter connections in real-time. On the smartphone, a manual proxy configuration is set up, with the local IPv4 address being used as the server address and 8080 as the port. The mitmproxy's Certificate Authority (CA) is then installed on the smartphone.

When an application establishes an HTTPS connection, it verifies the legitimacy of the server's certificate through comparison with the trusted system certificate authorities listed in the Android operating system. The list of CA is fixed and secure, but some applications may choose to implement their own custom certificate validation process, known as "Certificate Pinning". We bypass this process by using the Apk-mitm [103] tool. This is accomplished through the application of a series of steps, including 1) decoding the APK file with Apktool (more details in Section 4.6), 2) replacing the application's network security configuration to allow user-added certificates, 3) modifying the source code to disable various certificate pinning implementations, fourth, encoding the patched APK file with Apktool, and finally, 4) signing the patched APK file with uber-apk-signer [104]. The application of the apk-mitm to the extracted SumUp APK file results in the creation of a modified version of the app. This modified app now trusts the mitmproxy certificate, which is added to Android's built-in list of trusted system certificate authorities, allowing for the interception of traffic sent to SumUp's payment provider servers.

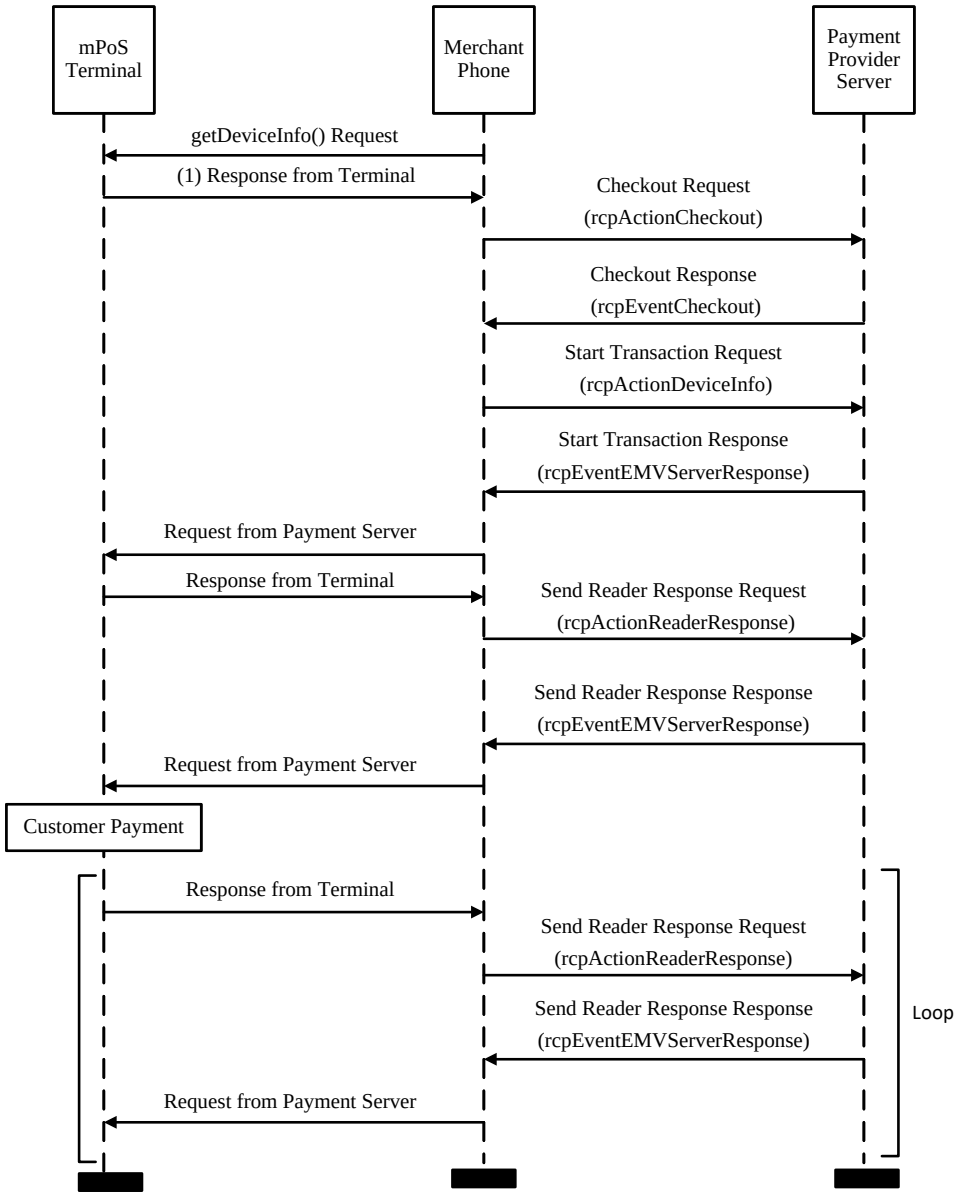


Figure 4.5: Sequence Diagram of the Exchanged Messages

Table 4.2: Exposed Commands in SumUp Application Source Code

Command Name	**Base64-Encoded Command**
PINPLUS DEVICE POWER OFF COMMAND	AAIBAQ4=
PINPLUS SHOW DEFAULT MESSAGE	ABUBAQsAAAABAAtTdW1VcCBQSU4rAP8A

4.5.2 Tampering Attack

In this proof-of-concept demonstration, we present a tampering attack that highlights the feasibility of data modification. In this scenario, a MITM attack is utilized to intercept and manipulate the communication transmitted during a transaction.

By tampering with the messages sent by the payment server for the terminal, we can change the behaviour of the terminal for fraudulent purposes. The messages from the payment server are commands that tell the terminal what to do next to proceed with a transaction. Aside from the messages that we see in network traffic analysis, there are two commands exposed in the application source code, as can be seen in Table 4.2. The PINPLUS SHOW DEFAULT MESSAGE command is used to show a default message of "SumUp PIN+" on the terminal's display. If we decode the command into hexadecimal, the command contains this string in plaintext ASCII. This means that we can insert arbitrary ASCII into this command to display arbitrary text on the terminal's display.

However, there are limitations to this attack. Protected messages cannot be altered, as the terminal will reject them, resulting in an error message. Additionally, unprotected messages are not accepted by the terminal during protected message exchange. This presents a problem as modification and sending of commands are desired during a transaction, which largely involves protected message exchanges. The "leave_protected_session" command, which is sent in response to the payment server during a protected message exchange, provides a solution. Tracing its usage in the source code as shown in Fig. 4.6, reveals its sole purpose is to end a protected message exchange in case of errors. This allows us to propose an attack on the SumUp terminal by exploiting the ability to exit a protected message exchange at any point during a transaction.

The ability to leave a protected message exchange at any point in a transaction allows us to propose an attack on the SumUp terminal. At the end of a

```
@Override
public void onError(i.t.n.a.c.b bar, @Nullable List<j> list, h hVar)
     {
    String str = "onError␣event␣received.␣error␣code:␣" + hVar;
    if ((hVar == i.t.n.a.d.b.NOT_ALLOWED ||
    hVar == i.t.n.a.d.b. INVALID_SEQUENCE_NUMBER_IN_PROTECTED _MODE
         && ReaderCoreManager
    ReaderCoreManager.this.leave_Protected_Mode();
    }

    else {
        WReaderModuleCoreState.getBus().m(new CardReaderErrorEvent(
            bar, ReaderCoreManager.this.isReadyToTransmit(), list));
    }
}
```

Figure 4.6: Usage of Leaving a Protected Session in the SumUp's Application Source Code

normal transaction, the payment server will send two commands to the terminal to inform it that the transaction was successful. In our attack, we replace these two commands to trick the terminal into displaying that the payment method was declined. First, we use the "leave_protected_session" command sent earlier in the transaction to exit the protected message exchange, allowing us to send an unprotected command. This is followed by the PINPLUS SHOW DEFAULT MESSAGE command that has been modified to display the text "Declined" on the terminal's display. The result of this attack is a successful transaction with the terminal displaying that the transaction was not successful. This is shown in Fig. 4.7. This vulnerability could be part of a social engineering attack and multiple transactions could be carried out. In this scenario, the victim, who is the user making a contactless transaction, can protect themselves by requesting a receipt. The generated receipt would be sent via the mobile phone, accurately indicating the accepted transaction.

4.6 Software Security

The security of mPoS terminals can be analyzed through the reverse engineering of their code. Reverse engineering refers to the systematic examination of the code of a software program to comprehend its functioning, identify its vulnerabilities, and potentially modify it. In this section, we demonstrate the viability of reverse engineering the code of mPoS terminals mobile applications.

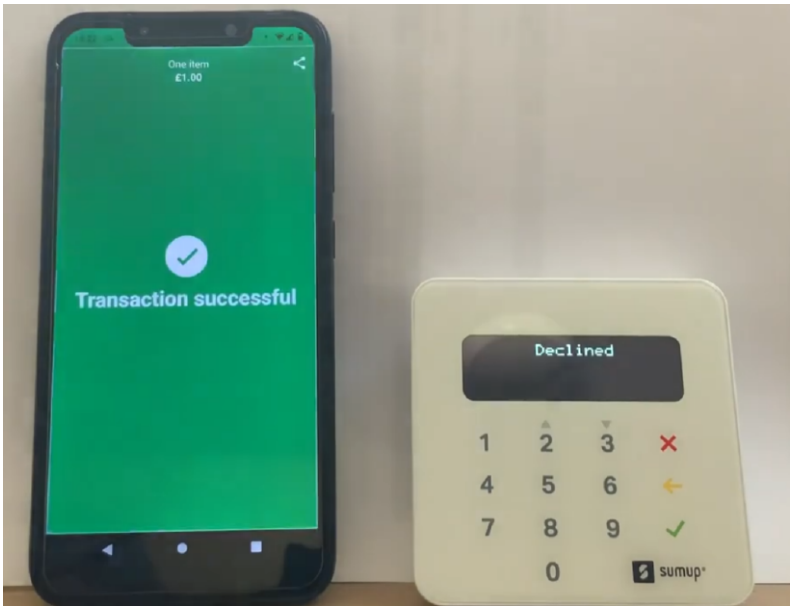


Figure 4.7: Tampering Attack on Transaction Messages

In particular, we employ an Android smartphone to analyze the source code and demonstrate the capability of modifying the behaviour of the mPoS terminal through the alteration of the mobile application code. In our case study, we use the SumUp Air mPoS terminal and the Android mobile application. To this end, we outline the procedures involved in the reverse engineering process and present the results of our case study. Our findings underscore the significance of adopting secure code development and deployment practices for mPoS technology to prevent potential security threats.

4.6.1 Reverse Engineering

The Android applications are primarily written in Java and are stored as Android packages in the Android Package Kit (APK) file format, which is essentially zip files that encompass resources and assembled Java code. The process of reverse engineering the APK files on Android phones includes several steps: de-compiling, making modifications, re-compiling, and signing the APK to be used on Android phones. We use the APK of the SumUp application and decompile using two methods, apktool [103] and a standard Java decompiler [27]. The first tool produces Smali code, while the second produces Java code. We use two different tools as they are complementary. Smali code is more

difficult to read, therefore we use Java code to understand the application code and identify the vulnerable parts that can be exploited, apply the changes in the relevant part of the Smali code and use it to rebuild and sign the code. To do this, we reverse the decompiling process by rebuilding and signing the APK. The APK was rebuilt using apk-mitm [119], which uses Apktool to encode the patched APK file and the uber-apk-signer [104] tool to sign and verify the APK.

4.6.2 Software Modification Attack

As outlined in Section 4.5.1, modification of the code can circumvent the Certificate Pinning mechanism, thereby allowing the attacker to execute MITM and tampering attacks on the communication between the merchant's mobile phone and the server of the service provider. Here, we demonstrate another software modification attack, showcasing how this vulnerability can be exploited to neutralize an additional security feature: *beep sound*.

The process of performing a contactless payment on an mPoS terminal is often accompanied by an audible beep sound as a security feature, which alerts the user to the transaction taking place. This serves as a notification to the user regarding the ongoing transaction and is essential in the prevention of relay attacks. However, a study of the SumUp Air card reader application showed that it is possible to compromise this security feature through modification of the app software.

The analysis of the code revealed that the volume of the beep sound is controlled by the *playSoundEffect* method within the *AudioManagers* class. By modifying this method, it is possible to completely control the sound and disable this security feature. In addition, the keyboard input sound made by the SumUp app can also be muted through modification of the code. This involved removing all function declarations and calls related to the *playSoundEffect* method from the code base. The recompilation and installation of the modified application showed that the sound is no longer played when keyboard inputs are used during the charge creation process. This highlights the vulnerability of the application to modification and raises concerns about the potential for malicious actors to manipulate the app and compromise the security protocols designed to protect customers. This finding underscores the importance of employing more secure solutions to ensure the safety of user transactions. Relying solely on an audible beep sound as a security feature is insufficient and poses a significant

risk to users.

4.7 Discussion

4.7.1 Ethical Disclosures

The present study was performed within a controlled setting. The authors purchased commercially available mPoS terminals and used their own bank accounts to demonstrate the proof-of-concept attacks. Our research primarily focused on the SumUp Air mPoS terminal. We have shared our findings with the vendor for their review and feedback. We are currently in discussions with them to further address these issues.

4.7.2 Mitigating the Vulnerabilities

During our study, we have identified possible solutions for the security issues of mPoS terminals, as explained below.

Secure Pairing Method: The use of secure pairing methods is crucial in enhancing the security of communication between mPoS terminals and payment servers. The presence of a display on the SumUp Air device highlights the possibility of adopting a more secure pairing method in order to safeguard the communication conducted via Bluetooth Low Energy (BLE) [120]. It is recommended that mPoS terminal providers implement Secure Connections pairing methods (such as Numeric Comparison) instead of LE Legacy pairing. This is because Secure Connections methods utilize elliptic-curve Diffie–Hellman algorithms to generate public/private key pairs and generate an LTK during Phase 2 of pairing instead of an STK. The adoption of secure pairing methods is especially important for mPoS terminals, as these devices generally have limited input/output capabilities and are therefore more susceptible to security threats.

Traffic Security: In order to counteract the threat posed by traffic interception, it is vital for mPoS terminal providers to adopt secure network practices. This can include obscuring techniques [19] for terminal commands within the source code and protecting any commands that transmit unstructured text to the terminal display. Regular software updates can also help to fix known vulnerabilities and ensure that the security of the system remains up-to-date.

Code Protection: The security of mPoS terminal applications can be enhanced by implementing measures to protect the code from reverse-engineering by attackers. Code obfuscation and anti-tampering (AT) techniques can make it more difficult for attackers to access and exploit vulnerabilities in the code. The code obfuscation techniques hide informative data in the software, making it hard to understand for both humans and decompilation tools [19] as well as protecting against repackaging [143]. The anti-tampering techniques allow the app to both detect alterations from its original state by checking the integrity of the code and to verify the source of the app itself (i.e. the app store where the app comes from) [19]. These techniques can prevent unauthorized access to sensitive information and mitigate the risks associated with code tampering.

4.8 Conclusion

This chapter analyzes the security implications of mobile Point-of-Sale (mPoS) terminals and their relationship with merchant's mobile phones as a key component of the mPoS system. The security aspects of communication between the (merchant's) mobile phone and the mPoS terminal, the mobile phone, and the payment server, and also the security risks in the mobile phone application itself are examined. An eavesdropping attack is performed to reveal cryptographic keys in the BLE communication, a man-in-the-middle (MITM) attack is performed to tamper with mPoS terminal messages, and the mobile phone application is reverse-engineered to alter the security features of the mPoS terminals controlled by the mobile phone.

Chapter 5

OPay Solution for Contactless Passive Relay Attacks

5.1 Overview

The usage of contactless payments has surged in recent years, especially during the COVID-19 pandemic. A Passive relay (PR) attack against a contactless card is a well-known threat, which has been extensively studied in the past, with many solutions available. However, with the mass deployment of mobile point-of-sale (mPoS) devices, there emerges a new threat, which we call mPoS-based passive (MP) attacks. In an MP attack, the various components required in a PR attack, including an NFC reader, a wireless link, a remote card emulator, and a remote payment terminal, are conveniently combined into one compact device; hence, the attack becomes much easier. Since the attacker and the victim are in the same location, the previous distance-bounding or ambient sensor-based solutions are no longer effective. In this chapter, we propose a new orientation-based payment solution called OPay. OPay builds on the observation that when a user makes a legitimate contactless payment, the card and the terminal surface are naturally aligned, but in an attack scenario, this situation is less likely to occur. This allows us to distinguish legitimate payments from passive attacks based on measuring the alignment of orientations. We build a concrete prototype using two Arduino boards embedded with NFC and motion sensors to act as a card and a payment terminal, respectively. To evaluate the feasibility, we recruited twenty volunteers for a user study. Participants generally find OPay easy to use, fast, and reliable. Experiments show that OPay can substantially reduce the attack success rate by 85-99%

with little inconvenience to real users. To the best of our knowledge, OPay is the first solution that can prevent both PR and MP attacks while preserving the existing usage model for contactless payment.

5.2 Introduction

It is well known that existing contactless cards are vulnerable to (passive) relay (PR) attacks, as previously discussed in Section 3.3.2. Due to the passive nature of contactless cards, anyone who is near the victim can launch this attack without the victim's awareness. The user may discover this attack later when they receive the bank statements, but the money has already been stolen. Such attacks can be difficult to trace, especially when the payments are made at unattended terminals, e.g., a self-service kiosk [118].

Passive attacks against contactless cards have become increasingly concerning in recent years for two reasons. First, the spending limit for a contactless payment has increased significantly. When contactless cards were first introduced in the UK in 2007, they were limited to only £10 in a transaction. However, this limit quickly rose to £20 in 2012, £30 in 2015, £45 in 2020, and £100 in 2021 [65]. With the increasing limit, contactless cards are becoming a more attractive target. Second, the number of mobile PoS (mPoS) terminals has been quickly growing (e.g. SumUp [133], Square [124], and iZettle [78]), as discussed in Chapter 4.

While mPoS devices bring great convenience to retailers and small businesses in setting up their payment terminals, they can also be easily misused. We use the SumUp device as an example. In our experiments, we entered an arbitrary amount under the spending limit on a SumUp device and were able to discretely deduct the amount from a user's card, which was kept in their bag or pocket. This proof-of-concept attack was tested against the cards of the authors, but the same attack can be trivially extended to steal money from anyone.

Currently, the primary countermeasure implemented in SumUp and other mPoS devices is making an audible "beep" sound when a payment is made. This serves to alert the card owner that a transaction has been made. However, in Chapter 4, Section 4.6.2, we showed how this beep sound can be muted by reverse engineering and software modification. A secondary countermeasure is to trace the bank account associated with the mPoS terminal and hopefully recover the stolen money. However, numerous examples of fraud in the banking industry suggest that recovering stolen money is not an easy task [5]. For example,

attackers may use mPoS terminals to wirelessly steal money from people in multiple crowded places like, train stations, shopping malls, or concerts, at the same time, so that they can steal a significant amount of money within a short period of time. They will simply withdraw or transfer out the money before being discovered. In reality, criminals often hire unsuspecting (young and old) people as mules and use their bank accounts as intermediaries to transfer illicit funds. All these make it difficult to trace the real attackers.

We consider an mPoS-based passive (MP) attack as a new form of passive attack. To some extent, an MP attack can be seen as a variant of a PR attack. A PR attack involves an NFC reader, a wireless link, a remote card emulator, and a remote terminal. In an MP attack, these different parts are conveniently combined into one compact mPoS device. This greatly reduces the sophistication of the equipment and skills required to carry out an attack.

As a result of this new variant of the passive attack, many solutions proposed in the past to defend against PR attacks are no longer effective. Common solutions in the literature are based on the assumption that the victim's card and the real terminal are far apart in two distinct environments. More concretely, they adopt distance-bounding protocols [29] or use sensors to measure the ambient environment (e.g., temperature [117], light [69], audio [69, 136], humidity [117], GPS [136], magnetic field [79] and infrared light [66, 68]) to ensure the two devices are in close proximity. However, in an MP attack, the fact that the card and the mPoS terminal are already in close proximity renders these solutions ineffective.

Besides the distance-bounding and ambient-sensor-based solutions, some researchers propose to prevent PR attacks by involving explicit user actions to activate the payment processes. For example, Tap-Tap and Pay (TTP) [95] requires a user to gently tap the card (or the mobile phone) against the terminal twice in succession to initiate a contactless payment. Shake on It (Shot) [132] requires the NFC card and the reader to be held together to establish physical contact via accelerators and vibrators. Proximity and Relay Attack Detection (PRAD) [67] works by requiring the user to press buttons on NFC devices to activate the transaction. While these solutions are useful in certain applications, they are less suitable in the context of contactless payment since they modify the usage model of how a user normally makes a contactless card payment.

To effectively prevent passive attacks against contactless cards, a practical solution should satisfy the following requirements:. First, it should prevent both PR and MP attacks, taking into account that the victim's card and the

real terminal may be in close proximity and in the same environment. Second, it should be fast, allowing the transaction to be completed within 500 ms according to the EMV requirement [46]. Third, it should preserve the usage model, allowing users to naturally complete a transaction as normal.

To the best of our knowledge, there is no existing solution that satisfies all of these requirements. Therefore, we present a solution that meets this goal. Without loss of generality, we focus on the more dangerous MP attack, but the same solution is also applicable to preventing the PR attack. The key idea in our solution is to make use of the accelerator and gyroscope sensors to derive the orientation of an NFC device. When a user makes a contactless payment by placing the card on the top or in front of an mPoS terminal, the orientations of the card and the terminal are naturally aligned. However, in an attack scenario where the victim's card is in a bag or pocket, the card and the terminal are less likely to be aligned. Hence, based on analyzing the orientations, we can tell a legitimate payment apart from an illegitimate one. We also build a concrete prototype and conduct a user study to evaluate the feasibility of our solution. The user study indicates that our solution is easy to use, and can substantially reduce the attack success rate from the current 100% to only 1-15% while incurring only a small 4.76% false rejection rate. We summarize our contributions as follows.

- We present OPay, an orientation-based payment solution against passive attacks in contactless payments. Our solution is the first that addresses both PR and MP attacks, supports a fast transaction under 500 ms and does not change the usage model.
- We build a concrete prototype of OPay by using Arduino boards with embedded NFC, accelerometer, and gyroscope sensors to implement a payment card and a terminal respectively. All our code is open source here.
- We conduct user studies to evaluate the usability and performance of our OPay prototype. The studies show that our solution is easy to use with low false positive and negative rates.

The rest of the chapter is organized as follows. In Section 5.3, we describe the threat model and the OPay system, followed by the system prototype and evaluation in Section 5.4. OPay is compared with related work in Section 5.5.

We finally discuss the limitations of OPay in Section 5.6 and conclude the chapter in Section 5.7.

5.3 Our Proposed OPay System

In this section, we propose an orientation-based payment system called OPay. The main idea of OPay is to use the orientation data of the payment device and the mPoS terminal in order to approve or deny a transaction based on the similarity of their measurements. The intuition is that when a user makes a contactless payment, the orientation of their card is naturally aligned with that of the payment terminal. In the case of an attack, when an attacker uses an mPoS terminal to approach an uncooperative user, it is less likely that the orientations of the two devices will be aligned. Our goal is not to completely stop the passive attacks but to significantly increase the chance of detection without adding inconvenience to users in legitimate payment scenarios.

5.3.1 Overview

Fig. 5.1 shows an overview of the architectural design of our system. In OPay, both the payment device and the mPoS terminal collect readings from the accelerometer and gyroscope sensors to independently calculate the orientations. The mPoS terminal sends a challenge to the card to initiate the NFC communication and to request a contactless payment. The card responds with signed transaction data, generated with a Message Authentication Code (MAC), e.g., using HMAC [129] and a MAC key k derived from the shared key between the card and the issuer bank. Then, the terminal forwards the transaction data to an issuer bank via a payment network. MAC protects the transaction data from being modified by the terminal or any entity in the transmission path. This follows the existing data flow in the EMV specification [26]. OPay does not change this flow but adds an encrypted blob of the card's orientation data, $\mathsf{Ori(c)}$, e.g., using AES-CBC [129] and a symmetric encryption key derived from the shared key between the card and the issuer bank [43]. The card's secret key shared with the bank is protected by the tamper-resistant chip, and hence cannot be accessed by the attacker (otherwise the bank cards can be cloned).

As discussed in Chapter 3, Radu et al. [109] demonstrated that Mastercard's Relay Protection Protocol (RRP), which is designed to prevent relay attacks, can be bypassed when used in different positions and angles and therefore lacks

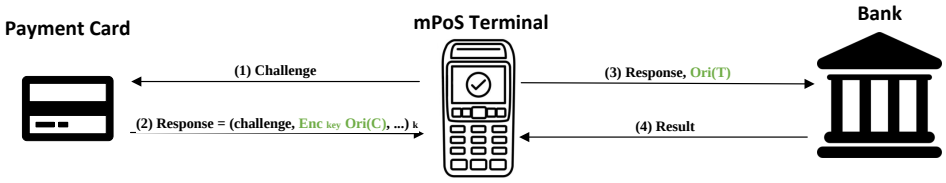


Figure 5.1: Architecture of OPay

robustness in the face of orientation changes. Utilizing the orientation data proposed in the OPay solution in the protocol data has the potential to address and improve upon these challenges.

As we will explain later, the orientation data consists of 4 float numbers (float-16), hence only 8 bytes. Accordingly, the mPoS terminal sends its own orientation measurement to the bank. If the difference between the two orientations is smaller than a threshold, the bank approves the transactions; otherwise, the transaction is denied, and the user needs to try again.

This solution preserves the existing usage model as a user makes a payment naturally as normal. It is important to note that, based on our observations, the orientation of the card and the reader tend to naturally align during contactless transactions made by our users on OPay. In the event that other users encounter any deviation from this behaviour when using OPay, they may need to position their cards at an aligned angle on the readers. While this may slightly differ from the conventional contactless payment method, the overall user experience aims to maintain a seamless and integrated process within the OPay solution. However, to an attacker, it raises the bar for a successful attack. Without OPay, a passive attacker can steal money with 100% success on the first attempt. However, with OPay, as we will show, while legitimate users can still normally make a successful payment on the first attempt, an attacker will need to make multiple attempts, which can significantly increase the chance of attack detection. For example, if the contactless payment fails consecutively three times due to the misalignment of the orientations, it will trigger an alert at the issuer bank, which in turn can send an SMS message to the user's phone to inform them of suspicious activity.

5.3.2 Threat Model

We consider an mPoS-based passive (MP) attack as the main threat. Compared to the PR attack, the attacker owns a PoS terminal and can carry out the

attack much more easily. Previous solutions to prevent PR attacks based on distance bounding and ambient environments no longer work since the card and the real terminal are actually in the same location during the MP attack. In our threat model, the mPoS terminal holder is malicious and aims to steal money from the user by getting close to their payment device. It is called passive because the attack can be done without the user's knowledge. The malicious terminal reads the victim's card passively to make a contactless transaction. The amount of the payment is variable up to the spending limit (now £100 in the UK). This attack can be performed in crowded places such as bus and train stations, a shopping mall, or a concert.

Random Guessing Attack: In this scenario, the attacker has no knowledge of the card's orientation, e.g., when the card is kept inside the user's bag. The attacker randomly chooses an orientation angle in the 3D space and rotates it until they succeed in aligning the two devices. In a random guessing attack, the attacker has a limited chance of success in each try and therefore needs to make several tries until the transaction is approved. Consecutively failed attempts will substantially increase the chance of detection by the bank.

Targeted Guessing Attack: We also consider the scenario in which the attacker has partial knowledge of the card's orientation, e.g., when the card is kept in a wallet in the user's pocket. Depending on the visibility of the pocket, the attacker knows that the orientation of the card may be limited to a certain range and hence has a higher chance of success in guessing the card's orientation. However, our solution still raises the bar for the attacker significantly. As opposed to merely approaching the victim's card within the NFC range (typically 10 cm) from any direction in any angle to make a contactless deduction, the attacker now needs to place the mPoS device near the victim's pocket with parallel alignment to the card's orientation. This significantly increases the chance of the attack being exposed to the user and nearby people.

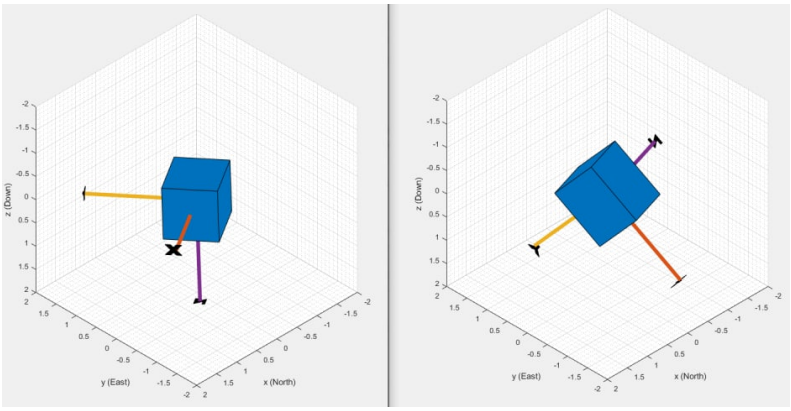
Attacks Beyond Scope: The malicious mPoS terminal holder may be equipped with a portable x-ray scanner and be able to see through opaque objects (e.g., bags) to analyze the orientation of the card. OPay is vulnerable to this kind of attack. However, the constant use of X-rays will present a health threat to the attacker, which can serve as a deterrent. It can also raise suspicion when used in public places. We note that certain cameras (e.g., OnePlus 8 Pro) claim to have an "x-ray vision", but they merely adjust the colour filter lens to let through infrared light, hence cannot see through opaque

objects as x-ray does [23]. OPay is also vulnerable to Denial-of-Service (DoS) attacks when an attacker intends to disrupt or manipulate the communication channel. As the malicious mPoS terminal holder intends to communicate with the payment device to steal money, they do not have the intention to disrupt the communication channel. Therefore, DoS attacks are out of the scope of this chapter.

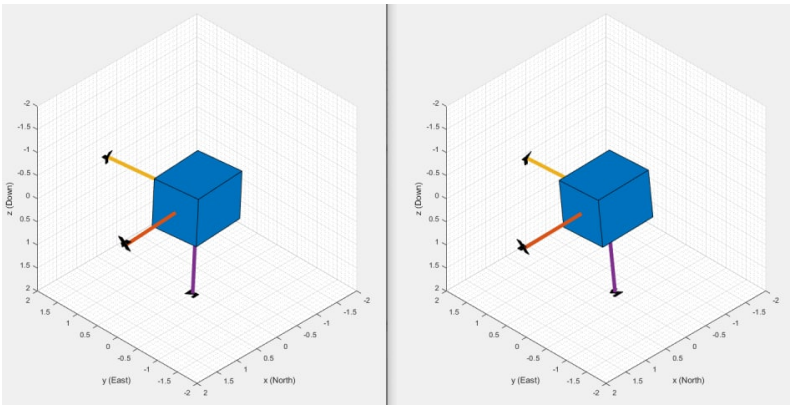
5.3.3 Orientation Estimation

For orientation estimation, three types of sensors are commonly used: accelerometer, gyroscope, and magnetometer. They measure acceleration, angular velocity, and local magnetic field, respectively. It is expected that combining all three sensors may give the best result. To verify whether this combination is suitable in the context of our application, we chose an MPU-9250 Multi-Chip Module (MCM) which has all these sensors. The MPU-9250 is a 9-axis Motion Tracking device that combines a 3-axis gyroscope, a 3-axis accelerometer, and a 3-axis magnetometer. In our prototype, this module was embedded in an Arduino board, and connected to a laptop for data collection. When we put the two Arduino boards together in close proximity to simulate a contactless payment process, we found fusing all three sensors gave a misalignment but fusing only the accelerometer and gyroscope data gave the expected alignment (see Figure 5.2). This is because when the two devices are placed in close proximity, the magnetometer measurements will be distorted due to the co-presence of a nearby magnetometer. Therefore, in our prototype, we only use the accelerometer and gyroscope data, which are fused by applying the six-axis Kalman filter algorithm [83] to estimate orientation.

We consider the definition of orientation as an angular displacement that can be described in terms of point or frame rotation. In point rotation, the coordinate system is static, and the point moves. In frame rotation, the point is static and the coordinate system moves. We use the latter to describe the orientation. Therefore, orientation is a rotation that takes a quantity from a parent reference frame to a child reference frame. We consider the geodetic coordinate system (earth) as the reference frame (parent), and the North-East-Down (NED) coordinate system as the coordinate frame (child), where the positive x-axis points north, the y-axis points east, and the z-axis points downward. To define three-dimensional frame rotation (axis of rotation), we rotate sequentially about the z, y, and x axes, respectively.



(a) Fusing Accelerometer, Gyroscope, and Magnetometer



(b) Fusing Accelerometer and Gyroscope

Figure 5.2: Orientation Alignments Between Two Aligned Devices

Orientation is usually represented as a quaternion, a rotation matrix, a set of Euler angles, or a rotation vector [83]. We use unit quaternions to represent orientation as they are more compact [28]. A quaternion is defined as a four-part hyper-complex number used in a four-dimensional vector space over the real numbers R^4. It is represented in the form of the following:

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \tag{5.1}$$

where a, b, c, and d are real numbers, and i, j, and k are the basis elements, satisfying the equation:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{5.2}$$

Every element of q has a unique representation based on a linear combination of the basis elements i, j, and k. We define an axis of rotation and an angle of rotation for each rotation (orientation) as below:

$$q = cos\ (\theta/2) + sin\ (\theta/2)(b\mathbf{i} + c\mathbf{j} + d\mathbf{k}) \tag{5.3}$$

where θ is the angle of rotation and $(b\mathbf{i} + c\mathbf{j} + d\mathbf{k})$ is the axis of rotation.

5.3.4 Similarity Comparison

There are multiple ways to measure distances between unit quaternions. Polar forms, dot product, and L_2 distance are the most popular forms [83]. Although these representations are in different forms, they are functionally equivalent. For simplicity, we choose the dot-product of the two quaternions for comparing and measuring the angle between them. Having the $q_t = a_t + b_t\mathbf{i} + c_t\mathbf{j} + d_t\mathbf{k}$ as the orientation of the mPoS terminal and $q_c = a_c + b_c\mathbf{i} + c_c\mathbf{j} + d_c\mathbf{k}$ representing the orientation of the card, the dot-product between them is defined as:

$$q_t \cdot q_c = a_t a_c + b_t b_c + c_t c_c + d_t d_c \tag{5.4}$$

The result of the dot-product is a scalar within the range $-1 \leqslant q_t \cdot q_c \leqslant +1$. Considering Equation (5.3) and using the absolute value of the dot product in Equation (5.4), we can calculate the angle (in the range of 0 and 90 degrees) between the two devices as follows.

$$\theta = \cos^{-1}(|q_t \cdot q_c|) \tag{5.5}$$

To show the correlation of the angle between the dot-product, we collected data for different orientation sets between the card and the terminal, with varying angles from 0 to 180 degrees. As one of the devices (the mPoS terminal) is fixed on the table, we rotated the other device (payment device/card) from 0 to 180 degrees. Fig. 5.3 shows the results where the x-axis is the degree of rotation and the y-axis is the dot-product in the range of 0 and 1. It can be seen from the diagram that the card and the terminal are in perfect alignment (i.e., $|q_t \cdot q_c| = 1$) when the angle is between 0 and 180 degrees and are perpendicular to each other (i.e., $|q_t \cdot q_c| = 0$) when the angle is 90 degrees. In our design, we consider the situation in which a user may make a transaction by either placing the front or back of their card on the PoS terminal. We treat them as

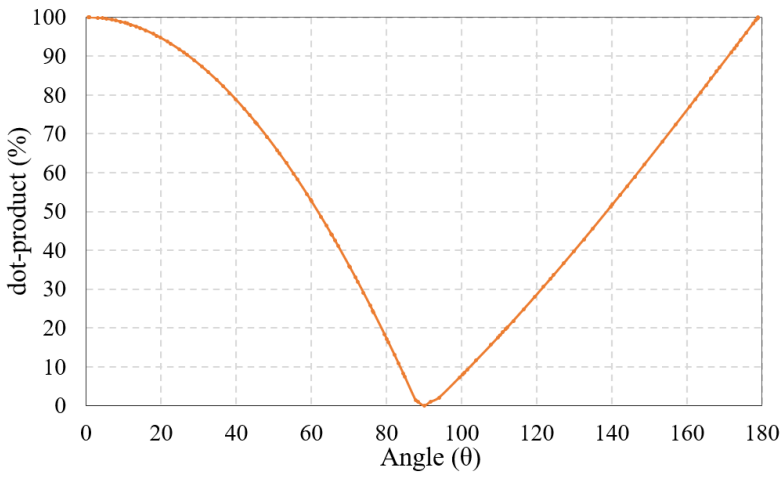


Figure 5.3: Correlation Between the Angle of Rotation and Dot-product of Quaternions

being equivalent, hence, the angles of 0 and 180 degrees are both considered aligned. In other applications, they can be treated differently if the user can distinguish the front and back of a card/device. In Figure 5.3, the values of the dot product are not completely symmetric according to the 90 degrees. This is because we embed the motion sensors on one side of the Arduino board, and the prototype of the card is not completely symmetric with reference to the board plane.

5.3.5 Threshold Calculation

To either accept or reject a transaction, the bank needs to make a decision based on comparing the orientation angles between the two devices. To calculate the threshold for the comparison, we use the False Acceptance Rate (FAR) and False Rejection Rate (FRR). FAR is the percentage of instances in which unauthorized transactions are incorrectly accepted. FRR is the percentage of instances in which authorized transactions are incorrectly rejected. The chosen threshold should give an appropriate trade-off between the security of the system and the usability experienced by users. In Section 5.4.3, we conduct a user study to determine the threshold and report the corresponding system performance.

5.4 System Prototype and Evaluation

We implemented a proof-of-concept prototype for the OPay system and conducted a user study to evaluate the system's performance.

5.4.1 Implementation

In the prototype, we developed two Arduino boards, one for the mPoS terminal and one for the card (payment device). On each of these boards, we used an MPU-9250 sensor for capturing the accelerometer and gyroscope data and a PN-532 NFC RFID module (version 3) for establishing the NFC communication between the two boards. Arduino Uno microcontrollers were used for programming these sensors. We used the P2P NFC communication between the two PN-532 modules in an Inter-integrated Circuit (I2C) mode, programming one NFC module as the initiator (acting as an mPoS terminal), and the other as the target (acting as a payment card).

When the user holds the card near the NFC field of the mPoS terminal to make a simulated contactless payment, the NFC sensor embedded in the terminal detects the presence of another NFC sensor in close proximity and hence initiates the NFC communication between the two devices. The motion sensors embedded on the two Arduino boards independently record the accelerometer and gyroscope measurements. In our proof-of-concept implementation, the collected sensor data on each board are transmitted via a serial port cable to a laptop for further processing. The orientations of the two Arduino boards, which represent the card and the terminal, respectively, are derived based on Section 5.3.3 and then compared. Based on the similarity, the transaction is either approved or rejected. The implemented prototype is shown in Fig. 5.4. In this prototype, the orientation data is derived from the accelerometer and gyroscope sensors.

5.4.2 User Study

Our user study involved 20 volunteers of different backgrounds from within and outside the university. Table 5.1 summarizes the demographics of the participants. The participant information leaflet can also be found in Appendix A.5. Our user study was ethically approved by our university's scientific research ethics committee. We also followed the UK government guidelines on COVID-19 to ensure the safety of our participants. While wearing face-covering during

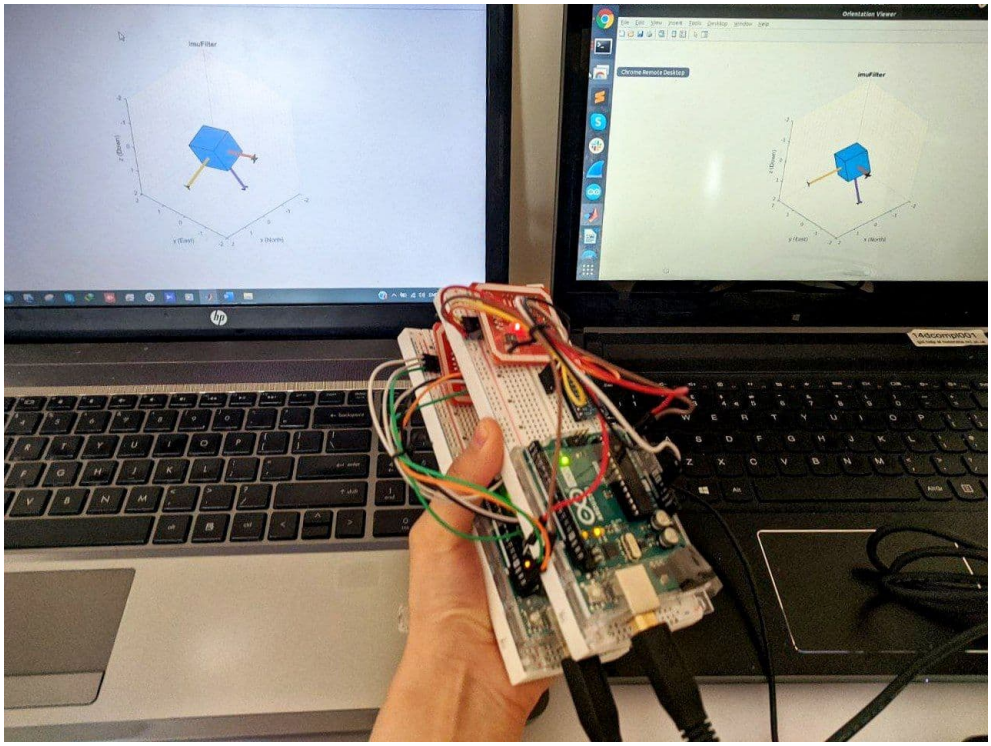


Figure 5.4: OPay Proposed Solution Prototype

all times of the study, we provided hand sanitizers, antibacterial wipes, and face masks to all of our participants and sanitized all surfaces after each user experiment.

In our user study, each of the participants performed three experiments, and in each experiment, the data collection was repeated five times. In the first experiment, we fixed the terminal's board on the table and asked users to hold the card's board to make a simulated contactless payment as they normally do in real life (see Fig. 5.5 a). In the second and third experiments, we asked the participants to act as attackers, considering the two attack settings: when the card's board is placed in a bag and when it is in a pocket. Fig. 5.5 b and Fig. 5.5 c show the in-bag and in-pocket attack scenarios, respectively. The same experiment was repeated five times. The recorded sensor data was saved into a file for further analysis.

Demographic	Participants(%)
Gender	
Male	12 (60%)
Female	8 (40%)
Age	
18-25	5 (25%)
26-35	9 (45%)
36-45	4 (20%)
46-55	2 (10%)
Occupation	
University Students	9 (45%)
University/Industry Employee	7 (35%)
Unemployed	4 (20%)

Table 5.1: OPay Participant Demographics (N=20)

Figure 5.5: User Study Setup: a) OPay Payment Setup; b) Random Guessing Attack; c) Targeted Guessing Attack

5.4.3 Performance

Error rates: as discussed in Section 5.3.5, we use FAR and FRR to evaluate the performance of OPay. The chosen angle influences system performance. A larger angle improves usability and lowers the FRR but it also raises the risk of FAR in attacks. Conversely, a smaller angle increases attack difficulty (low FAR) but may elevate the FRR and hinder usability. Future research on specific use cases is essential to identifying the optimal angle that balances usability and security. Fig. 5.6 shows the FRR and FAR results with reference to a threshold angle of varying degrees. For the targeted guessing attack, the equal error rate (EER) where the FRR and FAR curves intersect is 12%. For the random guessing attack, the EER is only 1%. As an example, if we choose $\theta = 5°$ as the threshold, we have FRR = 4.76%. For the targeted guessing attack,

FAR = 15.24%, and for the random guessing attack, FAR = 0.96%. This result is encouraging as it shows that we can substantially reduce the attack success rate from the current 100% to about 1-15% (that is a reduction by 85-99%). Hence, the attacker must make multiple tries, which will significantly increase the chance of detection by the issuer bank, which will in turn inform the user, e.g., by sending an SMS or a notification on the user's phone. The 4.76% false rejection rate is reasonably small. On average, the user will need to make $1/(1-4.76\%) = 1.05$ attempts to make a successful payment. This is hardly an inconvenience. In real-life contactless payment transactions, a cardholder is occasionally declined at the first attempt and needs to make a second attempt for the payment due to various reasons, e.g., distorted signals or interference with other nearby cards or NFC devices [54].

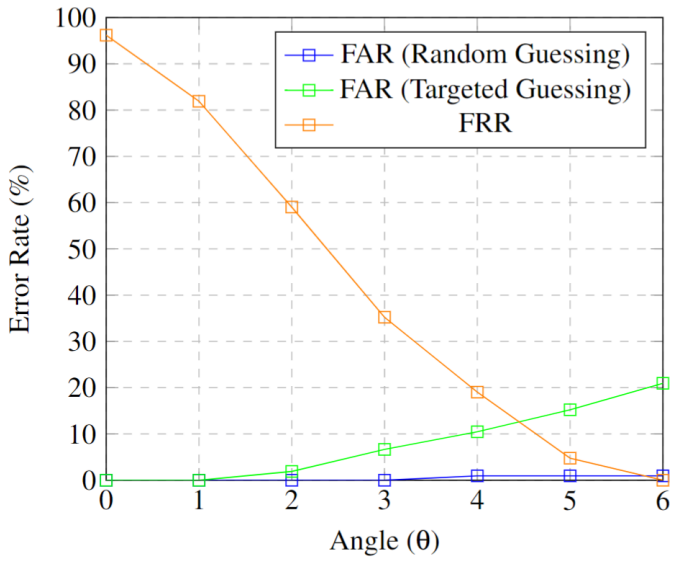


Figure 5.6: OPay Error Rates based on User Study

Timing: In terms of timing, our orientation detection requires collecting five samples of quaternions to derive the orientation of the device. It takes only 0.132 seconds to read data from the accelerometer and gyroscope sensors, as shown in Table 5.2. The remaining operations involve fusing the accelerometer and gyroscope measurements and calculating the orientation, which takes 0.082 and 0.014 seconds, respectively. Overall, the total duration is 0.228 seconds. From the user feedback, participants in our user study generally do not feel a difference in latency from a normal transaction. We note that providing a fast

payment experience is important, and EMV requires a contactless payment to be completed within 0.5 seconds.

Code	**Total Time (s)**	**% Time**
Read Sensor Data	0.132	58.1%
Sensor Data Fusion	0.082	36.2%
Orientation Calculation	0.014	5.7%
Total	**0.228**	**100%**

Table 5.2: Orientation Estimation Duration

5.4.4 Usability

After the experiments, we conducted an anonymous survey using a questionnaire. In the questionnaire, we asked our participants to rate both the normal contactless payment scenario and the OPay contactless payment scenario in terms of usability. We adopted a widely used System Usability Scale (SUS) framework to assess the user's satisfaction with usability [21]. The SUS questionnaire contains ten questions. The answer to each question scales from 1 to 5 (from strongly disagree to strongly agree). Table 5.3 shows the SUS questions along with the scores for both payment methods. The overall SUS score for the normal contactless payment scenario (without OPay) is 83. The score for the OPay contactless payment system is 78.62. The slight drop (5.28%) in the SUS score is mainly because the proof-of-concept prototype of the sensor-enabled card uses an Arduino board and is bulkier than a normal bank card. One user commented: "The prototype boards are heavy, and there are jumpers on them that make it difficult". Another user also commented: "I find it difficult for people with certain conditions, like people with Parkinson's or old people with shaking hands." Nonetheless, we are still encouraged by the SUS score of 78.62, which shows the user's general satisfaction with our prototype. We expect the SUS score to increase if the implementation of the card prototype can be made more compact.

In OPay, users make a contactless payment naturally as normal. The measurement of the motion sensor data is transparent and seamlessly integrated into the payment process. All these make users feel that the OPay system is as fast as a normal payment. A user commented: "To me, it is not different compared to the standard contactless payment scenario." The normal payment usage model is preserved as no additional action is required.

In the questionnaire, we also ask users about the frequency of using contact-

Questions	Average Rate without OPay	Average Rate with OPay	Questions	Average Rate without OPay	Average Rate with OPay
1. I think I would like to use this system frequently	4.25	4.45	2. I found the system unnecessarily complex	1.5	1.8
3. I thought the system was easy to use	4.52	4.5	4. I think that I would need the support of a technical person to be able to use this system	1.55	1.9
5. I found the various functions in this system were well integrated	4	4.15	6. I thought there was too much inconsistency in the system	1.9	1.85
7. I would imagine that most people would learn to use this system very quickly	4.55	3.85	8. I found the system very cumbersome to use	1.55	2.05
9. I felt very confident using this system	3.95	4.35	10. I need to learn a lot of things before I could get going with this system	1.55	1.75

Table 5.3: SUS Questions and Results

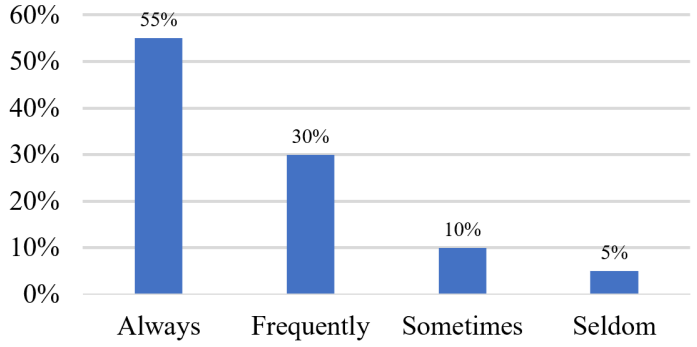
less payments in real life, among the choices of "always", "frequently", "sometimes" and "seldom". The majority of the participants (55%) chose "always", and 30% chose "frequently". Overall, most participants have had experience with using contactless payment (see Fig. 5.7a). By using the Spearman correlation method, we find a positive correlation between the OPay SUS score with the participant's previous experience of using contactless (see Fig.e 5.7b), i.e., the more experience of using contactless payment, the higher the SUS score (Spearman correlation coefficient $\rho = 0.301$ and two-tailed $p < 0.0001$). Similarly, as shown in Fig. 5.7b, there is also a positive correlation between the SUS score for a normal contactless payment system and the frequency of the usage ($\rho = 0.285$ and $p < 0.0001$).

5.5 Related Work

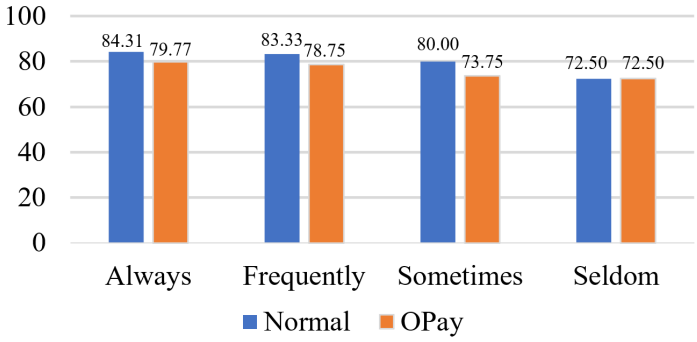
Contactless payment is one application of NFC technology for making an electronic payment. Other NFC applications include contactless access cards, keyless doors, keyless entry cars, etc. Passive relay (PR) attack is a common

Table 5.4: Comparing OPay with Other Solutions

Papers	**Type**	**Sensor(s)**	**Time (s)**	**FRR (%)**	**FAR (%)**	**Preserve Usage Model**	**Prevent Same Env. Attack**	**Feasible to Plastic Cards**	**Require EMV Improve**
Czeskis et al. [25]	User act.	Accelerometer	1	0	0	No	Yes	Yes	No
Gurulian et al. [67]	User act.	Force Sensitive Resistors	Seconds	0.1	0.1	No	Yes	No	Yes
Mehrnezhad et al. [95]	User act.	Accelerometer	0.6–1.5	9.99	9.99	No	Yes	Yes	Yes
Gurulian et al. [68]	Ambient env	Infrared Sensor	0.5	0.5	0.5	Yes	No	No	Yes
Gurulian et al. [66]	Ambient env	AAE Sensors	0.5	1.72	18.06	Yes	No	No	Yes
Ma et al. [87]	Ambient env	GPS	10	67.5	67.5	Yes	No	No	Yes
Halevi et al [69]	Ambient env	Audio	1-2	0	0	Yes	No	No	Yes
		light	1-2	5	6.5	Yes	No	No	Yes
Shrestha et al. [117]	Ambient env	Temperature (T)	Instant	23.74	32.40	Yes	No	No	Yes
		Gas (G)	Instant	15.26	30.36	Yes	No	No	Yes
		Humidity (H)	Instant	16.25	29.81	Yes	No	No	Yes
		Altitude (A)	Instant	8.57	16.25	Yes	No	No	Yes
		HA	Instant	7.93	9.85	Yes	No	No	Yes
		HGA	Instant	5.30	6.83	Yes	No	No	Yes
		THGA	Instant	2.96	5.81	Yes	No	No	Yes
OPay	Orientation	Accelerometer, Gyroscope	0.228	4.76	0.96-15.24	Yes	Yes	Yes	Yes



(a) Frequency of Using Contactless Payment



(b) Correlation with SUS Scores

Figure 5.7: Frequency of Contactless Payment Usage and Correlation with SUS Scores

threat to all these systems. Solutions proposed in the past can be generally divided into three categories: based on 1) distance bounding; 2) user activation and 3) ambient environment. For the specific contactless payment application discussed in this chapter, we focus on reviewing solutions in the last two categories. It is well known that distance bounding protocols are extremely sensitive to processing delays [123]. More efficient protocols apply symmetric cryptography but require the two devices to have a pre-shared secret key. This is not applicable in our scenario since the card and the payment terminal have no pre-shared secret. Furthermore, in an MP attack, the card and the terminal are already at a close distance. Hence, distance bounding is not applicable here.

User Activation: this category of solutions involves an explicit user action to activate the payment process. For example, Mehrnezhad et al. [95] proposed a "Tap-Tap and Pay" (TTP) solution, in which a user initiates an NFC payment by physically tapping their payment device against the reader

twice in succession to start the payment process. Czeskis et al. [25] require the user to perform a specific gesture (e.g. alpha, key/hip twist, single/double circle, and triangle) with their card to activate an authentication process. Their solution is designed for RFID access cards, but it can also be applied to prevent relay attacks in contactless payment. Gurulian et al. [67] require the user to press buttons on the user's payment device to activate a contactless payment process. All these solutions can prevent PR attacks and MP attacks since explicit user action is required. However, this changes the existing usage model in contactless payments.

Ambient Environment: this category of solutions uses sensors to measure the ambient environment to make sure the card and the reader are in the same environment or the same location. Halevi et al. [69] proposed to measure the audio and light in the ambient environment. Ma et al.[87] proposed to use the GPS data to ensure the card and the reader are in the same location. Shrestha et al. [117] proposed to measure the ambient environment using a range of sensors, including temperature (T), gas (G), humidity (H), and altitude (A). They further proposed to combine the sensors to improve results, e.g., GA which combines gas and altitude. Other combinations include HGA and THGA. Instead of measuring the natural environment, Gurulian et al. [66] proposed to use infrared light to create an Artificial Ambient Environment (AAE) and the infrared sensor to measure the environment. In a follow-up work [68], they proposed a similar solution of using vibration as an alternative AAE and six AAE sensors (accelerometer, gravity, gyroscope, linear acceleration, magnetic field, and rotation vector) to measure the surrounding environment.

While these ambient-sensors-based solutions can detect PR attacks when the card and remote terminal are located in two distinct environments, they have two limitations. First, the ambient environment is not a secret and can be easily manipulated as demonstrated by Truong et al. [136]. In an MP attack, the attacker has the freedom to manipulate the sounding environment of the mPoS device. For example, if the victim's card is kept in a bag and a light sensor is used to sense the ambient environment, the attacker can use a piece of clothing to wrap around the terminal to easily create the same dark ambient environment. Second, these solutions are generally designed for the scenario that the card and the reader are located in two remote locations with distinct environments, and therefore would not work when the devices are located in the same place, e.g., in an mPoS-based passive attack.

Comparison. OPay is a new orientation-based solution that does not

require explicit user action nor depends on the ambient environment. The user action involved in the payment is implicit and has been seamlessly integrated into a natural payment process. Therefore, it preserves the existing usage model. Table 5.4 compares OPay with related works. As compared to other solutions, OPay is reasonably fast, taking only 0.228 seconds in our prototype. The error rates (FRR = 4.76%, FAR = 0.96% for the random guessing attack, and FAR = 15.24% for the target guessing attack) present a reasonable trade-off in security and usability. It substantially reduces the chance of a successful attack with little inconvenience to users in a legitimate transaction. Some other works report better error rates than ours. However, we should highlight that a direct comparison of the error rates may not be appropriate since the test conditions are different. As an example, in Czeskis et al. [25], although the authors reported 0% FRR and 0% FAR, their user study involved only three participants, and all three participants were trained to practice a certain handshake before starting the experiments. In our user study, none of the twenty participants had any prior training on how to use OPay. They were asked to make a simulated contactless payment as they would normally do in a real-life transaction.

While the addition of certain sensors to plastic cards is deemed feasible (later discussed in Section 5.6, the overall viability of embedding diverse sensor types remains a subject of debate. Notably, ambient-based solutions detailed in the literature necessitate sensors such as gas, temperature, and GPS, which are arguably impractical for embedding in plastic credit cards. Furthermore, sensors related to user activation, such as Force Sensitive Resistors [67] or Infrared sensors [68], are specifically designed for touch screens, rather than being suitable for credit cards. Concerning modifications to the EMV protocol, all proposed solutions require enhancements to the protocol's messages, as the comparison of values is a critical aspect performed by a trusted third party, an entity typically shares a key with the card, which is usually the bank. An exception to this norm is found in the work of Czeskis et al [25], where a registration phase entails users performing a target action multiple times so that sensor data are recorded and serve as a template for comparing future actions. While the addition of the accelerometer sensor itself appears feasible for plastic cards, the practicability of incorporating storage for these templates requires further investigation by the industry.

It should also be noted that, if OPay is to be implemented in real-world scenarios, potentially serving as a solution to the discussed RRP protocol in

Section 5.3.1, the examination of reader coupling becomes imperative, especially in light of accommodating more variable angles. Factors influencing coupling include the relative size and positioning of antennas. In credit and debit card transactions, antenna size remains constant. However, a more in-depth analysis of positioning, specifically orientation data, is needed to better understand how card alignment affects coupling.

In general, ambient environment-based solutions preserve the existing usage model but are not effective when the attacker's device and the victim's card have the same or similar environment, or share the same location. Solutions based on user activation can prevent the same environment/location attacks but change the existing usage model. To our best knowledge, OPay is the first feasible solution that protects not only PR attacks but also MP attacks where the attacker is in the same environment or location as the victim while preserving the existing usage model.

5.6 Discussion

Feasibility of adding sensors: As shown in Table 5.4, using sensors is common in the proposed solutions to prevent passive attacks in contactless payments. The main research question pursued in this chapter is to identify which set of sensors we should use to prevent attacks without changing the existing usage model. We note that some commercialized bank cards have already been equipped with sensors, e.g., fingerprint sensors in Mastercard Biometric Card[1], which shows the feasibility of embedding sensors on bank cards. (However, note that the Master Biometric card requires the user to press the fingerprint sensor to make a payment, hence changing the existing usage model.) The prototype presented in this research utilizes an Arduino board and additional sensors, demonstrating the integration of sensor data with payment information. It is important to note that the prototype is not applied directly to physical credit cards but envisions the potential for sensor-embedded cards on the market. Our aspiration is that future credit cards may incorporate the same set of sensors employed in this study, facilitating the implementation of our proposed solution.

Usability: SUS is a widely used framework to assess users' satisfaction with the usability of computer systems [21]. It has been used in previous

[1]https://www.mastercard.us/en-us/business/overview/safety-and-security/authentication-services/biometrics/biometrics-card.html

studies [79, 82] to compare the usability among similar systems for pairing. We chose SUS over other usability tests such as Single Ease Question (SEQ) in order to establish a comparable benchmark for the usability of contactless payment systems. In our user study, we decided to use the original SUS questions without modification [21]. Users generally found the questions easy to understand. However, some users were puzzled by the words "inconsistency" in Q6 and "cumbersome" in Q8 (see Table 5.3), which shows a limitation of using SUS in our usability study. However, it is well-known that SUS questions are phrased for general purposes, and in a specific context, users may occasionally find the wording of some questions to not fit exactly [21].

5.7 Conclusion

In this chapter, we propose OPay, a novel orientation-based solution to prevent both passive replay attacks and mPoS-based passive attacks against contactless payment devices. We built a concrete prototype and conducted a user study to evaluate its feasibility. The users generally found our solution as easy to use as in a normal contactless payment experience; it was sufficiently fast, taking only 0.228 seconds; and it substantially reduced the attack success rate from the currently 100% to between 1-15% with only a small 4.76% false rejection rate. These make OPay a useful solution to fight against fraud in contactless payment systems.

Chapter 6

Users' Perception of Contactless Payment Security

6.1 Overview

In this chapter, we detail a user study with 150 participants from the UK, examining their perceptions of contactless payment systems and attacks. We explore their familiarity with the system and its adoption, their concerns and understanding of potential attack categories, and the protective steps they take. Conclusively, we compare users' perceptions with our evaluation of the technical feasibility of contactless payment attacks. We find that while users accurately interpret some attacks, they tend to overestimate certain attacks while underestimating others. In addition, in terms of protective actions, we find out that despite the availability of effective protective measures, users tend to employ only basic steps to safeguard their contactless payments from potential attacks. These findings highlight a gap between the user's perception of contactless payment attacks and their actual technical feasibility. We offer a set of recommendations, including enhancements to the security of contactless payment systems as well as education for users.

6.2 Introduction

While the technical aspects of contactless payment systems have been thoroughly researched, as explained in the previous chapters, the perspective of the end user, those who ultimately utilize these systems, remains relatively underexplored. This chapter aims to explore this pivotal aspect of the payment ecosystem

(refer to Fig. 1.1).Following the contactless payment attack categories presented in chapter 3, Section 3.3, we first evaluate the technical feasibility of these attacks. Second, we study users' familiarity and adaptation to contactless payment systems, their concerns and perceptions of the feasibility of these attacks, as well as the defensive measures they adopt to protect against such attacks. We then compare the users' perceived feasibility and concern regarding these attacks with our technical feasibility evaluation of attacks. Ultimately, this could strengthen the security and privacy of contactless payment systems. Essentially, this chapter is structured around the following research questions:

- RQ1: What is the technical feasibility of contactless payment systems attacks?
- RQ2: What are the users' perceptions and understandings of these attacks?
- RQ3: How well do users' perceptions align with the actual technical feasibility of these attacks?

To address RQ1, we first classify the various contactless payment attacks found in the literature based on their goal that impact users, as explained in Chapter 3, Section 3.3 and subsequently assess the technical feasibility of each of these categories based on our defined factors specified in Section 6.4. In response to RQ2, we conducted an in-depth study with 150 participants. This study explores user adoption patterns for contactless payment systems, their awareness of diverse attack types, their concerns and perceptions regarding the feasibility of these attacks, and the protective actions they deploy against such threats. Findings from this user study are showcased and analyzed in Section 6.6. For RQ3, we compare the users' perceptions and concerns about contactless payment attacks with our evaluation of the attacks' technical feasibility, further detailed in Section 6.7.1.

Our results show that users widely adopt contactless payment; meanwhile, they show varying levels of concern towards different attacks. While users have high concern levels for some attacks, they are less concerned by other ones. Users' concern level generally matches their perceptions of an attack's feasibility, with a few exceptions where, despite considering some attacks as less feasible, they still showed high concern about them. This suggests complexity in user perception and attitude. Despite these concerns, our results show that users typically take few protective actions, although multiple protective

actions are available to adopt. In order to close these gaps, we advocate better standardization and enforcement of contactless payment systems to improve their vulnerabilities, as well as user education and awareness.

The remainder of this chapter is organized as follows. In Section 6.3, we review the related work. Section 6.4 provides the technical feasibility of contactless payment attacks. The methodology and results of our user study are presented in Sections 6.5 and 6.6, respectively. This is followed by our discussion in Section 6.7, and finally, we conclude the chapter in Section 6.8.

6.3 Related Work

Contactless payment systems have been examined in the literature, spanning various perspectives, including factors driving their adoption, user experiences, and the impact of the COVID-19 pandemic. While research conducted in various countries has provided insights into these facets, it is important to acknowledge that the scope of understanding user's perceptions of contactless payment attacks, particularly in the UK context, is limited. This section discusses relevant, yet limited, user studies regarding payment systems conducted in different regions, emphasizing the need for further in-depth studies.

Studies exploring the adoption of contactless payment and mobile payment technologies during the COVID-19 pandemic have identified several influential factors. For instance, perceived risk emerged as a significant factor in shaping consumer behaviour in one study [72]. Another study conducted in Saudi Arabia highlighted the importance of health safety and hygiene considerations among consumers during the pandemic [116]. In India, studies have shown that perceived trust [2] and a combination of perceived risk and usefulness [139] play crucial roles in the adoption of mobile payment contactless technologies. In addition to the pandemic-related studies, there have been investigations into the intention to adopt mobile payments and e-wallets among consumers in different countries. Studies conducted in India [138], Pakistan [147], and Malaysia [62, 100] have examined factors influencing consumer adoption of mobile payment technologies. These studies have considered various factors, such as perceived usefulness, perceived ease of use, and trust.

The influence of culture and country-specific factors on user behaviour have been shown to be important factors in payment user studies. A research study [22] examined payment cultures in four countries, emphasizing the importance of considering these cultural differences. However, limited research has been

conducted in the context of the UK. Existing studies in the UK have focused on everyday spending behaviours and experiences [86], as well as the impact of cashless fares in the Transport for London (TFL) system [107]. One specific study in the UK explored people's mental models regarding risk perception associated with contactless debit cards [3]. This study addressed liability, severity, and likelihood perceptions related to card theft and fraud. However, the study had limitations in terms of the attack scenario, as it only presented a single situation involving the theft of a victim's purse for fraudulent contactless payments.

To the best of our knowledge, there is no dedicated user study conducted within the UK that comprehensively assesses users' perceptions of contactless payment attacks while also comparing these perceptions with the technical feasibility of such attacks. In this chapter, our primary objective is to bridge this gap by undertaking a comprehensive user study. We intend to present and analyze our findings in light of an extensive review of the existing contactless payment attacks.

6.4 Contactless Payment Attacks Technical Feasibility

As already explained in Chapter 3, Section 3.3, we have categorized contactless payment attacks into distinct categories including Data Leakage, Relay, Re-play, Card Replica, Limit Bypass, and Lock-screen Bypass. We intentionally exclude the Cryptogram Exploitation category, given that it either overlaps with the attacks within our six outlined categories or primarily targets merchants rather than users. Here, we evaluate the technical feasibility of each of these attack categories based on our defined factors.

We have identified several factors that can facilitate an analysis of the technical feasibility of each attack category. The *Replicability* of an attack is defined as the ability to repeat an attack and achieve the same results, falling into three categories: "fully replicable" if all attacks in a category are currently executable; "partially replicable" if some attacks have been mitigated but others persist; and "non-replicable" if all known methods of attack have been effectively addressed. In terms of *Affected Devices*, credit/debit cards and mobile phones are typically the main targets. While some attacks exclusively target cards, others focus on NFC-enabled mobile phones, and a subset can affect both types

of devices. The *Required Equipment* denotes the bare minimum hardware, excluding the victim's device and the payment terminal, needed to perform an attack. This can be improved to perform the attack with less equipment; however, we report the equipment that attackers deploy to perform the attack reported in their research. Certain attacks necessitate a *Compromised/Modified Terminal*, while other attacks will work with a regular terminal. It should be noted that compromising a terminal is considered a difficult and challenging task. The *Required Time* for an attack signifies the duration necessary to carry out the attack successfully, with some attacks being executed in real-time while others require additional time for preparation and execution. Lastly, the number of *Present Crook* refers to the necessity of having an attacker actively or passively involved in the attack, depending on the attack type and the threat model. The analysis of these factors for each attack category is shown in Table 6.1 and explained below.

Data Leakage is categorized as "partially replicable". Although encryption measures are in place, our experiments indicate that certain data can still be accessed from cards. An example of this leaked data based on our experiments can be found in Appendix A.1. Attacks reported in the literature [33, 71] primarily target cards and only require one NFC reader, without the need to compromise the terminal. This NFC reader could be conveniently positioned at a checkout counter, enabling data to be read in real-time without requiring an active crook, or data can be read with one active crook approaching a victim that has a card.

Relay attacks [20, 24, 30, 57, 70, 80] are fully "replicable" as relay attacks are still feasible, regardless of the relay protection measures, as shown in [109]. The log data of our recent relay attack, illustrating this, can be found in Appendix A.2. As previously mentioned, these attacks necessitate the use of two emulators in real time, indicating the need for two present crooks. The attacks aim at cards and can be executed without the user's knowledge or the need to compromise the terminal.

Pre-play attacks' replicability [52, 61, 110] cannot be fully determined at this time. However, considering that the attack from [61] remains replicable and the one from [110] is patched according to [18], we consider this type of attack "partially replicable". While other attacks in this category [52, 110] do not require a compromised terminal, the attack in [61] can affect both cards and phones when the terminal is compromised. The number of present crooks required can be one or none, depending on the threat model.

Card Replica attacks are also "partially replicable". While we cannot assess the feasibility of all attacks in this category [52, 59, 102], Visa's decision to remove the mag-stripe mode [42], affecting [52], and the continued replicability of the attack in [59] lead us to categorize this threat as "partially replicable". The recent replicable attack in 2019 [59] involves one active crook who reads data from two interfaces (EMV and magstripe) and later transcribes it onto a blank card, a process that requires time. These attacks can be executed without compromising the terminal.

Limit Bypass is considered "partially replicable", with [17, 31, 32] identified as patched attacks, while [16, 18, 61] are still replicable. These active threats can target both cards and phones, require two NFC readers, and can be executed without compromising terminals. The attack can occur in real-time and requires one present crook for the stolen payment device threat model, and two when the card is in possession of the victim and the attack happens in real-time.

Locks-screen Bypass attacks are also "partially replicable". They primarily target phones with lock-screen and necessitate two NFC readers and a laptop acting as a proxy server. Some scenarios, such as ApplePay-Visa bypass[1], do not require a compromised terminal and can happen in real-time, while others, like GooglePay-Mastercard, demand the terminal to be compromised and require time (50 attempts for a success rate of 22%). The number of crooks required is similar to Limit Bypass attack, depending on the threat model.

6.5 Methodology

In this section, we present the design of the survey, the data collection, and the analysis. This work has gained ethical approval from the ethics committee of the University of Warwick in the UK.

6.5.1 Survey Design

The design of the user study in this section is intentionally kept general for two primary reasons. Firstly, each category contains various attacks, differing in their threat models, methods, and exploited vulnerabilities. Hence, an overall description can cover the objective of all attacks in the same category. Secondly, we aim to avoid an overly technical narrative, ensuring that our participants grasp the overall attack concept without unnecessary complexity. To this end,

[1]The data log of our replication of this attack can be found in Appendix A.3.

Table 6.1: Feasibility Comparison of Contactless Payment Attacks

Attack Category	Replicability	Devices	Equipment	Compromised Terminal	Time	Crook (no.)
Data Leakage [33, 71]	partially replicable	Cards	1 NFC Reader	No	Real-time	0-1
Relay [20, 24, 30, 57, 70, 80]	Replicable	Cards	2 NFC Readers	No	Real-time	2
Pre-play [52, 61, 110]	partially replicable	Card, Phone	NFC Reader	Some Yes	Real-time	0-1
Card Replica [52, 59, 102]	partially replicable	Card	NFC Reader, USB Card Reader, Card Writer, Blank Cards	No	Needs time	1
Limit Bypass [16–18, 31, 32, 61]	partially replicable	Card, Phone	2 NFC Readers	No	Real-time	1-2
Lock-screen Bypass [109, 135, 144]	partially replicable	Phone	2 NFC Readers, Laptop	Some Yes	Need time or Real-time	1-2

example scenarios have been included for enhanced comprehension as well. The survey consists of the following main sections. Further details regarding each section can be found in Appendix A.6.

Introduction and Consent: This section marks the initiation of the survey and provides participants with a summary of the study's objectives and procedures. We ensure to clarify that participation is entirely voluntary and that the data collection is anonymous, following ethical guidelines. Then, participants are asked to give their informed consent before proceeding further.

General Knowledge and Preferences: In this section, we aim to understand the participant's technology usage patterns and preferences. Hence, participants are inquired about their familiarity with contactless payment, their usage frequency, the payment devices that they use for contactless payment, as well as their preferences, likes, and dislikes linked to this method of payment and their interest in new features provided by contactless technology (e.g., contactless payment cash withdrawal [15]).

Perception on Contactless Payment Security: This section first asks participants about their general perception of contactless payment security along with the security of payment devices. Next, participants are presented with descriptions and examples of the six attack categories. They are asked to assess the feasibility and their degree of concern for each attack type. The example scenarios can be found in Table 6.2. This section concludes with a

reassessment of the participants' general security concerns regarding contactless payments.

Protective Actions: This section aims to gauge the participants' proactive steps toward protecting against unauthorized payments. Questions are framed around their account monitoring habits in response to unauthorized transactions. Further, we ask participants to choose their preferred protective actions from a list of twelve options to counteract the vulnerabilities of contactless payment attacks.

Demographics: Participants are asked to provide demographic information, including age, gender, and the highest level of education.

Finally, the survey concludes with a feedback section. To ensure participant compensation, a unique completion code specific to this project was provided, enabling participants to claim their compensation on the Prolific [108] platform, which is our data collection platform and is explained in the following section.

Table 6.2: Attack Example Scenarios for Contactless Payment

Attacks	**Example Scenario**
Data Leakage	Imagine you're standing in the payment queue at a coffee shop, completely unaware that an attacker nearby or the malicious coffee shop owner may exploit the situation. Once you proceed to make a contactless payment, they could utilize a skimming device to gather your payment information, including the Primary Account Number (PAN) and expiry date.
Relay	Imagine you're waiting in a shop line, and an attacker in close proximity to you gain access to your card by using a phone to interact with it, hidden in a pocket or bag. This phone then relays the obtained data to a second device located in a jewellery shop in real-time, where a purchase is made using your card information.
Pre-play	Imagine you're at a store, all set to make a contactless payment with your card or smartphone. Little do you know that the payment terminal has been compromised by attackers. They intercept your payment information, which is then used later to conduct multiple fraudulent transactions. You may not notice anything suspicious during the legitimate transaction.
Card Replica	Imagine you are on a bus, and someone uncomfortably leans close to you. Alternatively, imagine being at a shop where the merchant insists on swiping your card, claiming they only accept magnetic stripe (mag-stripe) payments (on a terminal that is compromised). In all such cases, the necessary card data is collected through data interception, to be later encoded onto a counterfeit mag-stripe card.
Limit Bypass	Imagine you have lost your card. In this scenario, as criminals lack knowledge of your PIN, their only way to steal money from you is by utilizing your card for a contactless transaction up to the £100 limit in the UK. However, using the mentioned equipment, they can bypass this limit and make transactions of higher amounts, such as £1000 if available in your account.
Lock-screen Bypass	Imagine you are in a restaurant, and you leave your phone on the table unattended for a few seconds, or you are in a crowded place and put your phone in your bag, assuming that it is locked. The attacker gets fairly close to your locked phone, initiates a contactless payment with a terminal near it, and by using special equipment such as additional smartphones that run malicious codes, changes the payment information and convinces your phone that it is making a payment to a transit operator, so it does not need to unlock.

6.5.2 Data Collection and Analysis

We designed our survey utilizing Google Forms and ran a series of pilot studies to verify the comprehensibility and consistency of the attack descriptions, especially ensuring that technical terms were understandable to participants. The first pilot study asked feedback from five experts on the study's design, with subsequent second and third pilot studies focusing on testing the survey's clarity among a broader audience. These pilot studies facilitated the identification and rectification of minor errors, informed necessary structural adjustments based on received feedback, and provided a benchmark for the time required to complete the questionnaire. Data collection was streamlined through Prolific [108], an online platform dedicated to simplifying participant recruitment and management for research. This platform allowed us to recruit 150 UK-based participants. Details regarding participant demographics are shown in Table. 6.3.

Participants were compensated for their involvement. While our pilot studies suggested a survey completion time of 10 minutes, we allocated a generous 20-minute time slot to cater to slower respondents. In the actual data collection, the average completion time aligned with our pilot studies at approximately 10 minutes.

In the process of analyzing the collected data, we employed a range of techniques to facilitate a comprehensive understanding of the survey results. Our initial approach involved a descriptive analysis to examine the survey responses in each category, providing insights into technology usage patterns, concerns about attacks, and participants' protective actions. Through this analysis, we could generate an overview of the central tendencies within the data, understanding the common behaviours and perceptions amongst our diverse sample of participants. Furthermore, we implemented a pre-post analysis to evaluate how the presentation of detailed information about various types of contactless payment attacks influenced the participants' perceptions of security. This comparison between their initial views and their views post-familiarity with contactless payment attacks enabled us to gauge the impact of increased awareness of security concerns.

Demographic	Participants(%)
Gender	
Male	72 (48%)
Female	76 (50.7%)
Non-binary/Third gender	1 (0.7%)
Prefer not to say	1 (0.7%)
Age	
18-24	21 (14%)
25-34	52 (34.7%)
35-44	41 (27.3%)
45-54	15 (10%)
55-64	12 (8%)
65 years or older	8 (5.3%)
Prefer not to say	1 (0.7%)
Highest Level of Education	
High school diploma or equivalent	26 (17.3%)
Some college or associate degree	39 (26%)
Bachelor's degree	54 (36%)
Master's degree	24 (16%)
PhD or higher	6 (4%)
Prefer not to say	1 (0.7%)

Table 6.3: User Study Participant Demographics (N=150)

6.6 Results

In this section, we will discuss user study results on contactless payment technology awareness and adoption, users' perception of the security of this technology, and finally the protective actions that users take.

6.6.1 General Knowledge and Preferences

In our exploration of user understanding and acceptance of contactless technology, we found a clear distinction in levels of familiarity. Approximately half of the users claimed comprehensive knowledge of how the technology functions, whereas about 40% only generally understood how it works. Interestingly, 10% of participants were aware of contactless technology but could not articulate how it operates. When asked to define contactless technology, most respondents displayed a clear understanding. They accurately described it as a method of making payments without needing to physically insert a card into a terminal or enter a PIN. The common theme was using a card or phone to tap over the payment machine, indicating a proper grasp of contactless payments. Participants also acknowledged the role of technologies such as RFID (radio frequency identification) and NFC in facilitating contactless payments. A few were even able to identify digital wallets like Apple Pay and Google Pay, demonstrating

their comprehension of these services in the context of contactless payments. Some users correctly mentioned a transaction limit (e.g., £100 under UK rules), while others expressed uncertainty about the specifics of the technology or gave oversimplified responses.

Table 6.4: Contactless Payment Adoption

Usage Frequency		**Preferred Devices (multiple choice)**	
Several times a day	24.70%	Contactless credit or debit card	97.30%
One or two times a day	40.00%	Mobile contactless payment	66.70%
One or two times a week	30.70%	Wearable contactless payment	11.30%
One or two times a month	4.60%		
Never	0.00%		
Likes (multiple choice)		**Dislikes (multiple choice)**	
It's fast	95.30%	Concerns about security and privacy	63.30%
It's convenient	95.30%	Technical issues	30.70%
It's Secure	29.30%	Maximum payment cap	22.70%
Other	0.70%	Lack of familiarity with the technology	3.30%
		Other	0.70%

As for adoption, we found that the use of contactless payment methods is widespread, with 99.3% of respondents having used it within the past six months. As illustrated in Table 6.4, the usage frequency varied among users, with 40% employing it once or twice daily, 30.7% weekly, and 24.7% multiple times per day. When it comes to preferred contactless payment devices, credit or debit cards were the most preferred devices with a rate of 97.3%. Mobile phones followed at 66.7%, while wearable devices lagged behind, with just 11.3% of respondents using them for contactless payments. When asked about their likes and dislikes about this technology, they expressed appreciation for the speed and convenience of contactless payments but also expressed concerns about security, technical issues and dissatisfaction with the maximum payment cap. Despite these concerns, about 85% of users considered contactless features important for businesses. As for new technologies, nearly half of the users (42.7%) still showed interest in newer contactless technologies like contactless cash withdrawal, which allows users to withdraw money by simply tapping their card on the ATM (Automated Teller Machine), without needing to insert their card into the terminal. An example of such technology is Barclay's contactless cash withdrawal feature [15].

6.6.2 Perception on Contactless Payment Security

The results indicating users' perceptions of general security concerns, payment device security perception, their concerns and perceived feasibility regarding categories of attacks, and revisited general security concerns are described below.

General Security Concern: We first asked participants what they thought about the overall security of making contactless payments. Only a small group of the people (around 18%) said they were concerned or very concerned about this. In contrast, nearly half of the users (around 49%) said they weren't concerned or were only somewhat concerned. Interestingly, the biggest group (around 33%) among the five categories did not feel strongly either way. This means they were unsure about how secure it is to make payments in this way.

Payment Devices Security Perception: Our analysis of the security perceptions surrounding contactless payment devices such as credit/debit cards, mobile phones, and wearable technologies are shown in Fig. 6.1. While credit/debit cards and mobile phones are generally viewed as secure by a majority of participants, wearable devices lag behind in perceived security. For credit and debit cards, a majority of participants (67.4%) view them as either "very secure" (18.7%) or "secure" (48.7%). A small percentage of participants (5.3%) feel these cards are "not at all secure". Approximately 16.7% of respondents perceive them as "not secure" while the rest (10.7%) maintained a neutral stance on their security. When it comes to mobile phones, a slightly lower percentage of participants (64%) view them as secure with 24% saying they are "very secure" and 40% considering them "secure". The view that mobile phones are "not at all secure" is held by an identical 5.3% of the respondents as with credit/debit cards. However, fewer respondents (10%) see mobile phones as "not secure" compared to credit/debit cards. A significant 20.7% remain neutral about mobile phone security, which is nearly double the neutral response for credit/debit cards. Wearable devices are perceived as less secure overall, with 48.6% of respondents viewing them as either "very secure" (15.3%) or "secure" (33.3%). This represents a decrease compared to the perception of security for credit/debit cards and mobile phones. A slightly higher percentage (6.7%) consider these devices as "not at all secure", and 16% perceive them as "not secure". Interestingly, wearable devices have the highest percentage of neutral responses (28.7%) among the three technologies discussed.

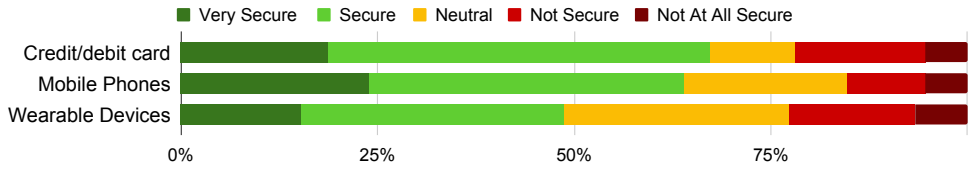


Figure 6.1: Users' Perception on Security of Contactless Payment Devices

Users Perceived Feasibility of Attacks: The feasibility of each attack from the user's perspective is depicted in Fig. 6.2. Our analysis indicates that Data Leakage and Pre-play attacks are regarded as the most feasible attacks from the users' perspective. For both these types, 54.7% of participants consider them "somewhat feasible", and 38.7% view them as "feasible". Only 6.7% of participants consider these attacks as "not feasible". This similarity in the results could be attributed to the similarity of the threat models in both attack scenarios; the attacker uses an NFC reader to approach the victim and steals data for fraudulent purposes.

On the other hand, Limit Bypass and Lock-screen Bypass attacks are perceived as less feasible than other attack types. For Limit Bypass attacks, 45.3% of respondents regard it as "somewhat feasible", 24.7% deem it "feasible", while a considerably larger group, 30%, considers it "not feasible" (highest "not feasible" compared to all attack categories). Similarly, for Lock-screen Bypass, 60% regard it as "somewhat feasible". 25.3% view it as "feasible", and 14.7% consider it "not feasible". These figures suggest a higher degree of uncertainty about the likelihood of these attacks relative to Data Leakage and Pre-play attacks.

Regarding Relay and Card Replica attacks, the "somewhat feasible" and "feasible" responses lie between the most feasible groups (Data Leakage and Pre-play) and the least feasible ones (Limit Bypass and Lock-screen Bypass). This suggests a moderate level of perceived feasibility among participants. For Card Replica attacks, 50.7% perceive it as somewhat feasible, 36% as feasible, and 13.3% regard it as not feasible. Meanwhile, for relay attacks, 58.7% consider them somewhat feasible, 32% regard them as feasible, and 9.3% deem them as not feasible.

Users Concern of Attacks: The results of users' concern level about each attack are shown in Fig. 6.3. We consider the "very concerned" and "concerned" categories as "high concern", and the "not concerned" and "not at all concerned" as "low concern". The survey results reveal that users exhibit varying degrees

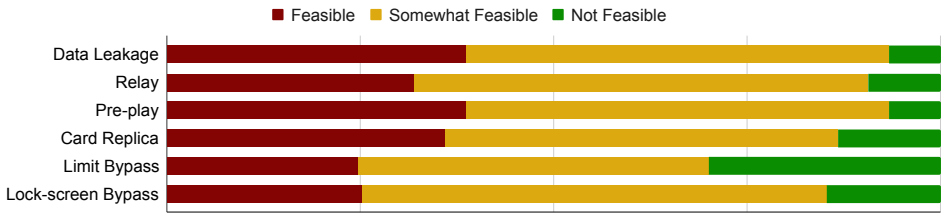


Figure 6.2: Users' Perceived Feasibility of Each Contactless Attack

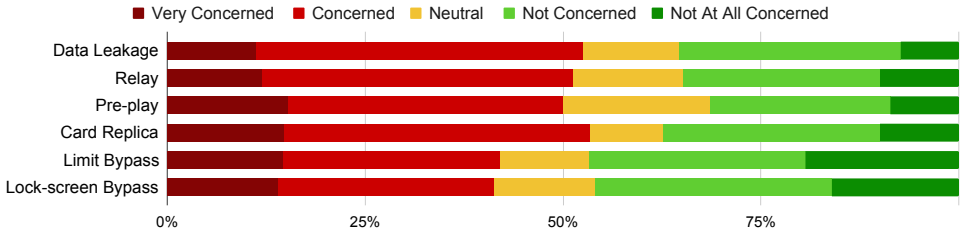


Figure 6.3: Users' Concerns Level on Different Contactless Payment Attacks

of concern regarding different types of attacks. For some types of attacks, such as Data Leakage, Relay, and Pre-play attacks, there is a relatively uniform level of concern. Approximately half of the users express high levels of concern about these three types of attacks, while around a third consider these attacks to be of lesser concern. Interestingly, users seem to exhibit a higher degree of uncertainty regarding the Pre-play attack. Approximately one-fifth of the respondents remain neutral about this type of attack, indicating a potential lack of understanding or familiarity with it. The Card Replica attack showed the most definitive reactions, with the lowest percentage of users remaining neutral among all types of attacks. The level of concern for this type of attack was the highest among all, with about 55% of respondents indicating high concern. Meanwhile, about 37% of respondents expressed low concern for the Card Replica attack. Finally, the concern levels for both the Limit Bypass and Lock-screen Bypass attacks were similar to each other but notably lower when compared with other categories of attacks. Only about 40% of the respondents expressed high concern for these two types of attacks, and almost half of the users (46%) had low concern. This significant deviation suggests a greater degree of uncertainty or possible underestimation of these attacks among users.

Comparing the concern levels with the perceived feasibility of attacks

generally reflects a pattern in the data; the perceived feasibility of an attack closely ties to the level of concern participants feel about it, with an exception. The Data Leakage and Pre-play attacks, which participants thought were the most feasible, also triggered a high level of concern. This alignment suggests that participants are more concerned about attacks they believe are most feasible. Similarly, the Limit Bypass and Lock-screen Bypass attacks which were viewed as the least feasible attacks, attracted less concern among users. However, the Card Replica and Relay attacks present an interesting deviation from this trend. Although participants considered them less feasible than the Data Leakage and Pre-play attacks, they still expressed substantial concern levels, similar to these attacks. This discrepancy is most pronounced for the Card Replica attack, which provoked the highest level of concern overall. This can suggest that despite the lower perceived feasibility, the potential consequences of these types of attacks are a significant source of worry for participants.

General Security Concern (Revisited): After providing the attack descriptions along with example scenarios and asking participants about the level of possibility and their concern about each attack, we again asked participants how they would evaluate the overall security of contactless payment. These changes can be seen in Fig. 6.4. We noted an increase in the proportion of users in the "very concerned" category, with the figure rising from 3.30% to 14%. Similarly, the "concerned" category saw an increase from 14.69% to 27.30%. This suggests that awareness of the threats inherent in contactless payment systems significantly increased the perceived level of concern among the users. Simultaneously, the "neutral" category saw a considerable decrease from 32.67% to 12.70%, suggesting that the information provided helped users form more definitive opinions regarding the security of contactless payment systems. Interestingly, the "not concerned" category remained relatively stable, shifting from 30.67% to 30%, indicating that for a segment of users, their concern level was not significantly influenced by the presented information. Finally, the proportion of users who were "not at all concerned" showed a slight decrease from 18.68% to 16.00%. This implies that even among those initially unconcerned, education had some impact in heightening their sense of concern.

6.6.3 Protective Actions

In response to the means of monitoring payment activities and accounts by users, a vast majority (92%) disclosed their use of mobile banking via their

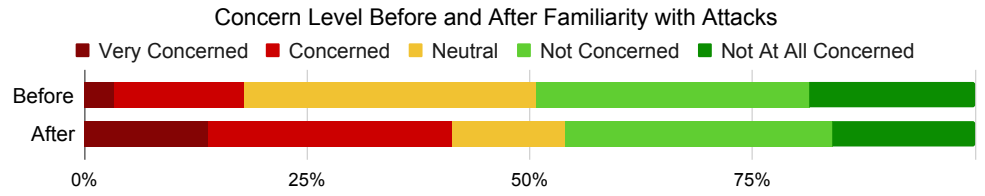


Figure 6.4: Participants Concern Level about Contactless Payment Before and After Familiarity with Attacks

smartphones, underscoring the popularity of this method. This was followed by 42% who opted for online banking on computers, whilst 20% leaned towards keeping and checking purchase receipts. These approaches were used either individually or in combination.

Results of the specific twelve protective actions that users take to protect their contactless payment security are depicted in Fig. 6.5. The most adopted security measure, as reported by 74% of participants, was regularly checking bank receipts and accounts. The second most taken protective action was being cautious when utilizing contactless payments in unfamiliar or untrusted environments, with 46% of respondents.

Approximately one in five (20%) participants choose not to carry cards or add cards with large amounts of funds to their digital wallets. This practice potentially mitigates attacks aimed at digitally pickpocketing from stolen cards. Close behind, 18% of participants have activated passive notifications such as SMS or calls for each transaction, while around 15.3% limit the total amount they can spend before their PIN is required. A few people (13%) adopt security measures such as RFID blocking wallets, disabling "express transit" mode on their digital wallets when not in use, or turning off the NFC sensor on their phones, all aimed at preventing unauthorized NFC access to their payment devices. Around 10% of respondents limit the maximum transaction amount and prefer reviewing monthly paper statements. The least common measures, as reported by about only 1% of participants involved disabling the contactless feature completely or requesting a card devoid of the contactless feature. Notably, about 10% of users do not implement any specific protective measures against contactless payment attacks.

The results reveal that despite the availability of various protective measures against the discussed attacks, users primarily rely on straightforward methods for protection. These include routinely checking their bank receipts and ac-

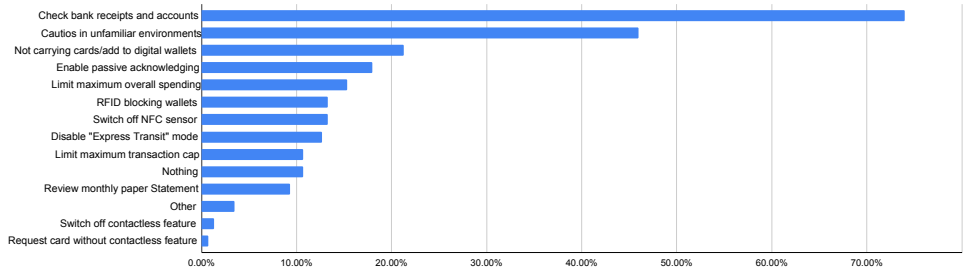


Figure 6.5: Protective Actions against Contactless Payment Threats Taken by Users

counts, and being cautious when utilizing contactless payments in untrusted environments. While these measures are beneficial, they may not be adequate against all threats. More advanced measures are used less frequently, suggesting a need for increased awareness and education on a wider range of protective measures.

6.6.4 Users' Feedback on the Survey

The feedback received from participants offers valuable insights about their experience with the survey. Many have expressed a positive view, stating that they found the survey both "straightforward and easy to understand" and "very informative and enjoyable". Several comments underscored the informative nature of the survey, with one participant noting "the number of possible modes of fraud is alarming". Others mentioned the educational value of the survey, saying, "I was not aware of all the ways my data could be stolen. I am much more aware now, thank you". A good number of participants appreciated how the survey made them rethink their security measures. One participant remarked, "This has really made me think about how little I do to protect myself from scams/attacks when using contactless". A recurring theme in the feedback was the newfound awareness and concern regarding the vulnerabilities of contactless payment. As one participant put it, "I had not really thought about contactless security before and did not know these methods of stealing data existed".

6.7 Discussion

In this section, we compare users' perception of the attacks with our evaluation of the technical feasibility of attacks, described in Section 6.4. We also discuss the limitations of our work along with recommendations for different stakeholders.

6.7.1 Users' Perception versus Technical Feasibility

Comparing the technical feasibility of various contactless payment attack categories in Table 6.1 with user perception in Section 6.6.2 reveals interesting insights about the users' understanding and awareness of potential threats. We can categorize the attack types into three categories: accurate estimation, overestimation, and underestimation of vulnerabilities.

Accurate Estimation: accurate estimation of vulnerabilities presents three key categories: Data Leakage, Relay, and Pre-play attacks. In the case of Data Leakage, users perceive this attack as highly likely and express significant concern, which aligns well with the technical feasibility. This accurate estimation of vulnerabilities might result from increased awareness about data privacy and security issues, fueled by frequent news about data breaches, leading to a realistic understanding of the vulnerabilities. Regarding Relay attacks, even though users were not sure about the feasibility of this attack, they still showed high levels of concern, which matches our technical feasibility evaluation. All the attacks in this category are still replicable, requiring only two NFC readers (typically phones) for real-time execution. This discrepancy points to the fact that while users recognize the vulnerabilities, they may lack full comprehension of the execution methods. Their limited awareness of these techniques, combined with the real-time nature and proximity requirement of these attacks, may contribute to the misconception that these attacks are less feasible than they truly are. Lastly, Pre-play attacks are perceived by users as highly probable, eliciting considerable concern. Their perception aligns with its technical feasibility, suggesting that users' understanding of these attacks is relatively accurate. Their comprehension of Pre-play attacks could be attributed to the intuitive nature of these attacks; the concept of an attacker intercepting and replaying transaction information might be easy to grasp, leading to an accurate estimation of its feasibility.

Overestimation: Users have expressed substantial concern about Card Replica attacks (more than any other category). However, this attack requires extensive time and specialized tools and is limited in its scope. Additionally,

the success of these attacks primarily relies on the use of mag-stripe mode, which has its limitations. The selling of card data becomes necessary for the attack to be fully effective, particularly as mag-stripe is now restricted in many regions [42]. This additional step increases the complexity of the attack, making it even less likely to happen frequently. This combination of factors contributes to a disparity between the perceived feasibility and the technical feasibility. It suggests that users may be overestimating this attack, likely fueled by concern over the potentially severe consequences of such attacks.

Underestimation: users tend to significantly underestimate Limit Bypass and Lock-screen Bypass attacks. In the case of the Limit Bypass attack, users have shown low concern levels compared to other attacks, and they have recognized this attack as the least feasible attack among all, regardless of the technical feasibility of these attacks and the high negative impacts that they have. This underestimation could stem from a lack of awareness of potential loopholes in transaction limits and EMV messages, and the fact that attackers can alter certain transaction data that are not authenticated by the bank. Similarly, for Lock-screen Bypass attacks, users regard these attacks as less likely and express less concern, indicating an underestimation of this attack, although it affects several digital wallets and several card brands. Users may underestimate the vulnerabilities of this attack due to faith in the security measures of their mobile devices, as also shown in Fig. 6.1 on the security of payment devices, particularly the lock-screen feature. The lack of familiarity with the "express transit" mode, used in some of these attacks, might also contribute to this underestimation, as about 85% of users reported never using this technology when asked in our survey (refer to Section 3.3.6 for more details).

While there is alignment between users' perceptions and the technical feasibility of attacks, there are clear gaps where attack vulnerabilities are either overestimated or underestimated. Overestimation, while a cautious approach, is vastly different from underestimation, which could lead to serious harm, exposing users to threats they are not fully aware of. This gap emphasizes the critical need for precise and easy-to-understand user education, helping users gain a realistic understanding of the vulnerabilities tied to contactless payment systems. By bridging this gap, we can increase users' awareness and empower them to protect against contactless payment attacks.

6.7.2 Limitations

While users have generally found the survey easy to understand, as shown in Section 6.6.4, we still consider the probability that some participants may have struggled to fully understand the technical aspects of contactless payment attacks due to the technical nature of these systems. In addition, our focus on the analysis of attacks documented in academic or peer-reviewed publications means we consciously excluded attacks reported in news outlets due to their often unavailable details. Hence, there is a possibility that a wider range of attacks exists in the wild. Furthermore, we only studied the users in the UK which was necessary since payment systems and attack vectors can differ across countries.

6.7.3 Recommendations

While users have found contactless payments fast and convenient, they have shown serious security concerns regarding contactless payment systems. To address these concerns, first, the security of payment systems needs to be enhanced, and second, users need to be educated about potential vulnerabilities and what they can do to protect themselves during these improvements.

For the first one, payment providers like Visa [142] and Mastercard [92], and standardization bodies such as EMVCo [37] should refine their protocols and introduce extra security checks to mitigate these threats. Furthermore, payment providers like Apple Pay [6], Google Pay [64], and Samsung Pay [114] should integrate additional protective features and ensure the security of edge devices, specifically NFC-enabled mobile phones. These phones are often the target of numerous attacks, however, while some users perceive them to be secure, they generally were more neutral regarding the security of mobile phones compared to credit/debit cards (as shown in Fig. 6.1).

For the second one, end users, who are the most important stakeholders in this ecosystem, must remain alert. They should strive to understand the vulnerabilities associated with contactless payments and actively take protective actions, as provided in Fig. 6.5, such as setting up and monitoring passive acknowledgement of payments, applying possible limits, regularly checking bank statements, and being careful when using contactless payments in unfamiliar or untrusted environments. Banks should also play their part in educating users about the variety of attacks that exist and the protective actions they can take on their online and mobile banking systems. Our results (Table 6.4) demonstrate

how education on contactless payment attacks can effectively increase users' level of concern about these vulnerabilities in payment systems. This is further supported by the feedback section results of our survey (discussed in Section 6.6.4), where the heightened awareness of contactless payment attacks and the potential protection methods made users rethink their security measures.

While the primary responsibility lies with the stakeholders to reduce vulnerabilities associated with payment systems, it's important to acknowledge that these systems can be vulnerable in various ways. As these vulnerabilities are addressed and security is improved, new vulnerabilities might surface. Therefore, it's unrealistic to claim that payment systems can be 100% secure. Consequently, it's beneficial for users to stay aware and updated about these vulnerabilities and to apply the recommended protective actions diligently.

6.8 Conclusion

Our comprehensive exploration of contactless payment attacks and their technical feasibility within the UK has revealed important insights into users' perceptions and understanding. Despite the ubiquity of contactless payment technology and its adoption in everyday life, users' perceptions of potential threats do not necessarily align with their technical feasibility. Users overestimate the vulnerabilities of Card Replica attacks due to fear of significant consequences, yet they underestimate Limit Bypass and Lock-screen Bypass attacks, likely due to unawareness of these methods and overconfidence in mobile device security. However, users accurately assess vulnerabilities associated with Data Leakage, Relay, and Pre-play attacks, indicating a better awareness in these areas. Our research also unveiled that to protect against these attacks, users often adopt basic measures such as reviewing their bank statements and being cautious in unfamiliar environments. Although these methods are effective, they are not sufficient to fully mitigate the potential vulnerabilities. These findings highlight the urgent need for improvement of payment systems vulnerabilities as well as increased awareness and education about the security vulnerabilities inherent to contactless payments and the potential protective actions.

Chapter 7

Conclusion

7.1 Summary

The primary objective of this research was to delve deep into the security of Card Present (CP) contactless payments across four angles: systematization and protocol analysis, attacks and vulnerabilities, countermeasures and solutions, and users' perspectives. We offered four critical contributions: 1) in terms of systematization and protocol analysis, we provided a comprehensive systematization of contactless payment attacks and the failures in their protocols, 2) in terms of attacks and vulnerabilities, we analyzed the security weaknesses of mobile Point-of-Sale (mPoS) terminals and showed how these terminals can be vulnerable in different ways, 3) in terms of countermeasures and solutions, we proposed a novel orientation-based contactless payment method, OPay, to prevent against relay attacks, specifically mPoS-based passive (MP) ones and 4) in terms of users' perspectives, we conducted a user study and compared users' perceptions of contactless payment security with our technical feasibility evaluation of attacks.

In Chapter 1, we began by painting a big picture of the payment ecosystem and delved into its core components: users, merchants, issuers, acquirers, and the payment network. Subsequently, we discussed the workings of two EMV payment systems, Card Present (CP) and Card Not Present (CNP) transactions, and their associated technologies.

Chapter 2 shifted the focus to contactless payment protocols, detailing the ISO 14443 standard used for card or object identification and the EMV contactless protocols. This includes the Entry Point, Kernel 2 (Mastercard), Kernel 3 (Visa), and the newly introduced Kernel 8, which was established as

a singular contactless kernel to simplify complicated multi-kernel systems.

In Chapter 3, we classified contactless payment attacks into seven distinct categories, namely Data Leakage, Relay, Pre-play, Counterfeit Card Replica, Contactless Limit Bypass, Lock-screen Bypass, and Cryptogram Exploitation. These are characterized based on their target protocol level: card-centric, cardholder-centric, or transaction-centric. Our analysis contrasting Visa (Kernel 3) with Mastercard (Kernel 2) revealed that Visa is more prone to cardholder-centric attacks, whereas Mastercard exhibits greater vulnerability to transaction-centric assaults. We then mapped these vulnerabilities to the affected protocols in Chapter 2, and discussed failures in the payment systems in different layers including Offline Mode, Mag-stripe Mode, Unencrypted Data, Unauthenticated Data, Unauthenticated/Compromised Terminal, and Ineffective Relay Protection and suggested potential countermeasures.

In Chapter 4, we analyzed the security of mPoS terminals, which accept contactless payment in a convenient way, from a different perspective, with a focus on the vulnerabilities of the merchant's phone. We showed how these devices are vulnerable in three different layers, including the communication between the merchant's mobile phone and the mPoS terminal, the communication between the merchant's mobile phone and the payment server, and the mobile phone application itself, installed on the merchant's mobile phone. We performed an eavesdropping attack on the communication between the mobile phone and the mPoS terminal to reveal the cryptographic keys in the BLE communication, performed a man-in-the-middle (MITM) attack on the communication between the merchant's mobile phone and the payment server to tamper with mPoS terminal messages, and finally reverse-engineered the mobile phone application to alter the security features of the mPoS terminals controlled by the mobile phone.

In Chapter 5, we focused on the Relay attack category as elaborated in Chapter 3. We demonstrate how the mPoS terminals, previously addressed in Chapter 4, can be exploited to digitally pickpocket from users, terming this method the mPoS-based passive (MP) relay attack. To make this attack as hard as possible, we proposed OPay, an orientation-based contactless payment solution, which is based on the observation that when a user makes a legitimate contactless payment, the card and the terminal surface are naturally aligned, but in an attack scenario, this situation is less likely to occur. Our solution is fast, taking only 0.228 seconds, is usable, with the SUS score of 78.62, reduces the attack success rate from the current 100% to between 1-15% depending on

the threat model, and does not change the usage model.

Chapter 6 presented a comprehensive user study undertaken in the UK, aiming to study users' perspectives as an important entity in the payment ecosystem as described in Chapter 1 and their perception of contactless payment system security. Our comprehensive exploration of contactless payment attacks based on our categorization in Chapter 3 and our evaluation of their technical feasibility has revealed important insights. Our findings indicated a disparity between users' perceptions and the technical feasibility of attacks. Results indicate that while users accurately estimate the vulnerability of certain attack categories, they tend to underestimate some attacks and overestimate others. Furthermore, our study on protective actions against these attacks shows that while effective protective measures exist, users often adopt basic ones, emphasizing the need for enhanced awareness and education about potential threats and protective measures.

In conclusion, we analyzed contactless payment systems from multiple angles, emphasizing the importance of a holistic view when addressing system security. Our findings highlight persistent research gaps and vulnerabilities within contactless payment systems across different levels.

7.2 Future Work

Future work is suggested as follows:

- **Chapter 3**: To enable an understanding of the dynamics between different types of attacks and countermeasures, it's crucial to introduce a standardized testing procedure. This procedure should offer a systematic approach for evaluating varied attacks across distinct card systems. The methodology should explicitly detail the testing parameters, environments, and card types. Additionally, it should provide specific criteria for assessing the risks of each attack, paving the way for more accurate assessments and consistent replication of results. To this end, the DREAD risk assessment model can be employed to quantify the level of risk associated with each attack category. This involves evaluating factors such as Damage, Reproducibility, Exploitability, Affected users, and Discoverability, and assigning scores to different aspects of the threat.

- **Chapter 4**: The scope for future inquiries extends to analyzing other mPoS terminals regarding their security loopholes. There is also a need

to not just identify these vulnerabilities but also explore potential countermeasures. These steps will contribute to a more comprehensive understanding of the security landscape of mPoS terminals and aid in the development of effective security measures to mitigate the vulnerabilities. We also plan to study the potential solutions further and evaluate the feasibility and effectiveness of these countermeasures in addressing the identified security issues.

- **Chapter 5**: Future research in this chapter includes investigating the feasibility of using OPay for wearable payment devices such as NFC-enabled wearable devices that are vulnerable to both PR and MP attacks. Applying OPay to these devices requires some adaptation of the definition of orientation for each device as the usage model varies with different payment devices.

- **Chapter 6**: Future research work includes expanding our study to other CP payment methods such as chip-and-PIN, QR code, and tap-and-PIN across multiple regions and deepening our understanding of payment risks and users' perceptions. We also plan to strengthen our findings with larger, more varied studies and explore the underlying causes of attack-related concerns. Furthermore, we aim to employ comparative and correlational analysis to understand the demographic impact on concerns and perceptions. Lastly, we plan to compare findings from different countries and add focus groups and interviews to our online surveys for a more thorough user perception analysis.

7.3 Research Directions

In the context of CP contactless transactions in payment systems, we consider the following as potential research directions:

The evolution of payment systems has brought forth the next generation of acceptance terminals, notably Tap-to-Phone [141] or Tap-to-Pay [89]. This groundbreaking technology incorporates NFC, facilitating merchants to accept contactless payments through their mobile devices without the need for an external terminal. Representing a notable evolution from traditional PoS and mPoS terminals, it offers merchants an efficient and cost-effective alternative to processing card payments. Tap-to-phone technology can be perceived as a new extension of the mPoS terminals designed for contactless transactions

that empower merchants to manage card payments using their mobile devices, simultaneously providing the convenience of accepting contactless payments directly. However, as with any technological advancement, there is potential for security vulnerabilities. These challenges and risks present intriguing topics for future research.

A deep dive into the realm of contactless technologies showcases that there are additional emerging solutions that need further research. Noteworthy among them are contactless cash withdrawal [15], and innovations like Visa and Mastercard's Pay-at-Pump systems [88, 140]. These advancements in the payment industry could have potential vulnerabilities and be interesting topics of study in the future.

Moreover, as we pivot towards a more interconnected and technologically advanced society, there is a surge in the adoption of NFC-enabled wearable devices. Devices like smartwatches (e.g., Apple Watch [9] and Samsung Galaxy Watch [113]), fitness trackers (e.g., Fitbit [55]), and unique innovations like smart jewellery (e.g., McLear Ring [93] and Kerv Ring [81]) that are able to make contactless payments are getting popular. These devices provide convenience, mobility, and modern fashion. However, the potential vulnerabilities of these payment-enabled devices remain ambiguous. Existing literature provides limited insight into their security challenges, and it's uncertain whether solutions proposed for other payment devices such as credit and debit cards and NFC-enabled mobile phones would seamlessly apply to these NFC-enabled wearable devices. This gap necessitates comprehensive research and analysis.

Finally, the introduction of the new EMV single kernel, Kernel 8, is a topic of interest. Given that the protocol demands more time for implementation in the wild, its comprehensive security analysis becomes paramount. This includes rigorous formal verification at the protocol level and a thorough assessment of potential vulnerabilities concerning attacks. The Kernel 8, while addressing complications presented by multi-kernel systems, might introduce its own set of unique challenges. Identifying and addressing these vulnerabilities is vital to ensure a robust and secure payment ecosystem, especially given that Kernel 8 appears to be the next era of contactless payment protocols.

Appendix A

Appendices

A.1 Data Leakage Attack Logs

As shown in Listing. A.1, using NFC reader tools such as NFC Reader [115], we can capture the card data, specifically Track 2 Data that contains Primary Account Number (PAN) and Expiration Date, specified in red. By using other tools such as Pro Credit Card Reader NFC [98], the card's transaction history can also be captured, if provided by the card.

Listing A.1: Data Leakage Attack Log

```
nfc.tag.id: 61292288
nfc.tag.tech: IsoDep, NFCA
card.aid: A0000000031010
card.pan:  4659xxxx6011 //Leaked Card.PAN
card.label: Visa Debit
card.tags.cm: FFI-Signature FFI-CVV2 FFI-Holog
* Send SELECT (PPSE) Command
+ Candidate AID: A0000xxxx1010 (Visa Debit)

* Send SELECT (A0000xxxx1010) Command
* Kernel "3" supported.
* Send GPO Command
! EP Outcome: "Card Read Complete"
! EP Message: "17" // Card read OK. Remove card
! EP Status: "Card Read Successfully"

cvn: "12" // Cryptogram Version Number (CVN) // 18
x57: "4659xxxxxxxx6011D26099201xxxxxxxxx001F"
// PAN and ExpiaryDate (09/26)//
```

```
X5F2D: "656E" // Language Preference // en
x5F34: "00" // Application PAN Sequence Number
x82: "2020" // Application Interchange Profile (AIP)
x84: "A0000xxxx1010" // Dedicated File (DF) Name
x87: "02" // Application Priority Indicator
X9F0A: "0001xxxxxxxx0000" // Application Selection Registered
    Proprietary Data (ASRPD)
X9F10: "060C1203A00000" // Issuer Application Data
X9F27: "80" // Cryptogram Information Data (CID)
X9F36: "000E" // Application Transaction Counter (ATC) // 14
X9F38: "9F66049F02069F03069Fxxx3704" // PDOL
C3: // EMV Contactless C-3, Visa PayWave tags
X9F6C: "1000" // Card Transaction Qualifiers (CTQ)
X9F6E: "20700000" // Form Factor Indicator (FFI)
```

A.2 Relay Attack Logs

As demonstrated in Listing. A.2, and by using the codes in [109], by employing a card emulator, a terminal emulator, and an NFC proxy server as an intermediary, it is feasible to relay payment data.

Listing A.2: Relay Attack Log

```

Received response from card emulator:
00A404000E325041592E5359532E444446303100
SELECT 2PAY.SYS.DDF01
Sending 00A404000E325041592E5359532E444446303100
Received command from terminal emulator:
6F2B840E325041592E5359532E4444463031A519BF0C1661144F07A0000000031
0109F0A0800010501000000009000
Status code: 9000 Command successfully executed (OK).
  6F | len:2B    File Control Information (FCI) Template
      84 | len:14    DF Name: 325041592E5359532E4444463031
      A5 | len:19    Proprietary Information
        BF0C | len:16    File Control Information (FCI) Issuer
            Discretionary Data
              61 | len:14    Directory Entry
                  4F | len:7    Application Identifier (AID):
                      A0000000031010
                9F0A | len:8    Application Selection Registered
                    Proprietary Data list: 0001050100000000
Sending
```

```
6F2B840E325041592E5359532E4444463031A519BF0C1661144F07A0000000031
0109F0A0800010501000000009000
Received response from card emulator:
00A4040007A000000003101000
SELECT A0000000031010
Sending 00A4040007A000000003101000
Received command from terminal emulator:
6F578407A0000000031010A54C500A5649534120444542495487010 29F38189F
66049F02069F03069F1A0295055F2A029A039C019F37045F2D02656EBF0C1A9F
5A0531082608269F0A0800010501000000000BF6304DF2001809000
Status code: 9000 Command successfully executed (OK).
  6F | len:57    File Control Information (FCI) Template
      84 | len:7    DF Name: A0000000031010
      A5 | len:4C    Proprietary Information
          50 | len:10    Application Label: 56495341204445424954
          87 | len:1    Application Priority Indicator: 02
        9F38 | len:18    Processing Options Data Object List (
            PDOL)
            9F66 | len:04   Terminal Transaction Qualifier (TTQ)
            9F02 | len:06   Amount, Authorised (Numeric)
            9F03 | len:06   Amount, Other (Numeric)
            9F1A | len:02   Terminal Country Code
              95 | len:05   Terminal Verification Results
            5F2A | len:02   Transaction Currency Code
              9A | len:03   Transaction Date
              9C | len:01   Transaction Type
            9F37 | len:04   Unpredictable Number
        5F2D | len:2    Language Preference: 656E
        BF0C | len:1A    File Control Information (FCI) Issuer
            Discretionary Data
            9F5A | len:5    Application Program Identifier:
                3108260826
            9F0A | len:8    Application Selection Registered
                Proprietary Data list: 0001050100000000
            BF63 | len:4    Unknown Payment System Tag: DF200180
Sending
6F578407A0000000031010A54C500A5649534120444542495487010 29F38189F6
6049F02069F03069F1A0295055F2A029A039C019F37045F2D02656EBF0C1A9F5A
0531082608269F0A0800010501000000000BF6304DF2001809000
Received response from card emulator:
80A8000023832136A040000000000001000000000000000826000000
0000082622030700 48D1F8B100
GPO command:
 9F66 | len  4    Terminal Transaction Qualifier (TTQ): 36A04000
    EMV Mode supported (Byte 1 Bit 6)
```

```
    EMV contact chip supported (Byte 1 Bit 5)
    Online PIN supported (Byte 1 Bit 3)
    Signature supported (Byte 1 Bit 2)
    Online cryptogram required (Byte 2 Bit 8)
    Contact chip offline pin supported (Byte 2 Bit 6)
    Mobile device functionality supported (Byte 3 Bit 7)
 9F02 | len  6    Amount, Authorised (Numeric) :000000000100
 9F03 | len  6    Amount, Other (Numeric) :000000000000
 9F1A | len  2    Terminal Country Code: 0826
   95 | len  5    Terminal Verification Results: 0000000000
 5F2A | len  2    Transaction Currency Code: 0826
   9A | len  3    Transaction Date: 220307
   9C | len  1    Transaction Type: 00
 9F37 | len  4    Unpredictable Number: 48D1F8B1
Sending
80A8000023832136A040000000000001000000000000000826000000000008262
203070048D1F8B100
Received command from terminal emulator:
7747820220005713XXXXXXXXXXXXXXXXD22112018500000000000F5F3401019F1
00706020A03A000009F260874D9D8E31871798F9F2701809F360201169F6C0216
009F6E04207000009000
Status code: 9000 Command successfully executed (OK).
  77 | len:47    Response Message Template Format 2
      82 | len:2    Application Interchange Profile: 2000
                    DDA supported (Byte 1 Bit 6)
      57 | len:19    Track 2 Equivalent Data:
          XXXXXXXXXXXXXXXXD22112018500000000000F
    5F34 | len:1    Application Primary Account Number (PAN)
        Sequence Number: 01
    9F10 | len:7    Issuer Application Data (IAD): 06020A03A00000
    9F26 | len:8    Application Cryptogram: 74D9D8E31871798F
    9F27 | len:1    Cryptogram Information Data: 80
    9F36 | len:2    Application Transaction Counter: 0116
    9F6C | len:2    Card Transaction Qualifiers (CTQ): 1600
                    Switch interface if offline data auth fails
                        and reader suports VIS (Byte 1 Bit 5)
                    Switch interface for cash (Byte 1 Bit 3)
                    Switch interface for cashback (Byte 1 Bit 2)
    9F6E | len:4    Form Factor Indicator (qVSDC): 20700000
Sending
7747820220005713XXXXXXXXXXXXXXXXD22112018500000000000F5F3401019F1
00706020A03A000009F260874D9D8E31871798F9F2701809F360201169F6C0216
009F6E04207000009000
```

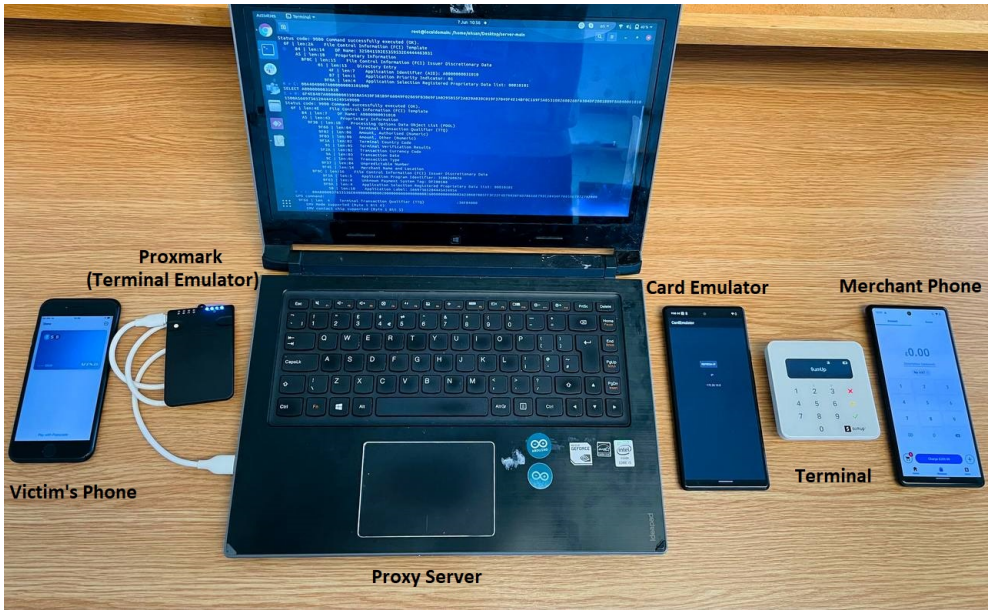


Figure A.1: Lock-screen Bypass Attack Demonstration Setup

A.3 Lock-screen Bypass Attack Logs

The successful transaction trace for the Lock-screen Bypass attack, using the code in [109], can be found in Listing A.3. Fig. A.1 shows our attack setup. In this setup, we first send the "magic string" to act as the "express transit" operator, and then change the Terminal Transaction Qualifier (TTQ) value, Byte 1 Bit 1, "Offline Data Authentication for Online Authorization" from zero to one, as shown in Table. A.1.

Listing A.3: ApplePay-Visa Lockscreen Bypass Attack Log

```
Sending the Magic String:  6a02XXXXXXXXXXXXXXXXXXXX00c2d8
// Transport for London (TFL) Data
R > C: 00A404000E325041592E5359532E444446303100
SELECT 2PAY.SYS.DDF01
026F2A840E325041592E5359532E4444463031A518BF0C1561134F07A00000000
3101087010 19F0A04000101019000
Status code: 9000 Command successfully executed (OK).
  6F | len:2A    File Control Information (FCI) Template
      84 | len:14    DF Name: 325041592E5359532E4444463031
      A5 | len:18    Proprietary Information
        BF0C | len:15    File Control Information (FCI) Issuer
            Discretionary Data
              61 | len:13    Directory Entry
```

```
                  4F | len:7    Application Identifier (AID):
                      A0000000031010
                  87 | len:1    Application Priority Indicator:
                      01
               9F0A | len:4    Application Selection Registered
                    Proprietary Data list: 00010101
R > C: 00A4040007A000000003101000
SELECT A0000000031010
036F428407A0000000031010A5379F381B9F66049F02069F03069F1A0295055F2
A029A039C019F37049F4E14BF0C169F5A053108260826BF6304DF2001809F0A04
000101019000
Status code: 9000 Command successfully executed (OK).
  6F | len:42    File Control Information (FCI) Template
      84 | len:7    DF Name: A0000000031010
      A5 | len:37    Proprietary Information
        9F38 | len:1B    Processing Options Data Object List (
            PDOL)
            9F66 | len:04   Terminal Transaction Qualifier (TTQ)
            9F02 | len:06   Amount, Authorised (Numeric)
            9F03 | len:06   Amount, Other (Numeric)
            9F1A | len:02   Terminal Country Code
              95 | len:05   Terminal Verification Results
            5F2A | len:02   Transaction Currency Code
              9A | len:03   Transaction Date
              9C | len:01   Transaction Type
            9F37 | len:04   Unpredictable Number
            9F4E | len:14   Merchant Name and Location
        BF0C | len:16    File Control Information (FCI) Issuer
            Discretionary Data
            9F5A | len:5    Application Program Identifier:
                3108260826
            BF63 | len:4    Unknown Payment System Tag: DF200180
            9F0A | len:4    Application Selection Registered
                Proprietary Data list: 00010101
R > C:
80A8000037833536A040000000000001000000000000000826000000000008262
21109009B07992E4D79436F6D70616E792C20436F76656E7472792000
GPO command:
 9F66 | len  4  TTQ :36A04000 //Old TTQ Value
    EMV Mode supported (Byte 1 Bit 6)
    EMV contact chip supported (Byte 1 Bit 5)
    Online PIN supported (Byte 1 Bit 3)
    Signature supported (Byte 1 Bit 2)
    Online cryptogram required (Byte 2 Bit 8)
    Contact chip offline pin supported (Byte 2 Bit 6)
```

```
    Mobile device functionality supported (Byte 3 Bit 7)
 9F02 | len  6    Amount, Authorised (Numeric): 000000000100
 9F03 | len  6    Amount, Other (Numeric): 000000000000
 9F1A | len  2    Terminal Country Code: 0826
   95 | len  5    Terminal Verification Results: 0000000000
 5F2A | len  2    Transaction Currency Cod: 0826
   9A | len  3    Transaction Date: 221109
   9C | len  1    Transaction Type: 00
 9F37 | len  4    Unpredictable Number: 9B07992E
 9F4E | len 20    Merchant Name and Location :4
    D79436F6D70616E792C20436F76656E74727920
... new TTQ: 23004000 //New TTQ Value
Status code: 9000 Command successfully executed (OK).
  77 | len:62    Response Message Template Format 2
      82 | len:2    Application Interchange Profile: 2040
                    DDA supported (Byte 1 Bit 6)
                    Expresspay Mobile supported (Byte 2 Bit 7)
      94 | len:4    Application File Locator: 18010100
                    SFI: 03, 1st record: 01, last record: 01, no
                        offline auth: 00
    9F36 | len:2    Application Transaction Counter: 0024
    9F26 | len:8    Application Cryptogram: B5FC8281477D36C7
    9F10 | len:32    Issuer Application Data (IAD):
1F426360A000000000100302730000000040000000000000000000000000000
    9F6C | len:2    Card Transaction Qualifiers (CTQ): 0000
      57 | len:19    Track 2 Equivalent Data:
         XXXXXXXXXXXXXXXXD23122017150099999995F
    9F6E | len:4    Form Factor Indicator (qVSDC): 23880000
    9F27 | len:1    Cryptogram Information Data: 80
R > C: 00B2011C00
READ RECORD: 01, SFI: 03
0370375F280208269F0702C0009F19060400100302735F3401009F241DXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX9000
Status code: 9000 Command successfully executed (OK).
  70 | len:37    Record Template
    5F28 | len:2    Issuer Country Code: 0826
    9F07 | len:2    Application Usage Control: C000
    9F19 | len:6    Token Requestor ID: 040010030273
    5F34 | len:1    Application Primary Account Number (PAN)
        Sequence Number: 00
    9F24 | len:29    Payment Account Reference (PAR):
        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Table A.1: Comparison of Terminal Transaction Qualifiers (TTQ)

Byte	1								2				3			4
Bit	1	2	3	4	5	6	7	8	1-5	6	7	8	1-6	7	8	1-8
Fields	**Offline Data Authentication for Online Authorization**	Signature	Online PIN	Offline-only / online	EMV contact chip	EMV mode	RFU	Mag-stripe mode	RFU	Offline PIN	CVM	Online cryptogram	RFU	Consumer Device CVM	Issuer Update Processing	RFU
Old TTQ (36E04000)	**0**	1	1	0	1	1	0	0	0	1	1	1	0	1	0	0
New TTQ (23004000)	**1**	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0

A.4 Cryptogram Confusion Attack Log (Failed)

Listing A.4 shows the data log for replicating the Cryptogram Confusion attack to pay with locked cards, as in [146]. We used a Visa card issued by Lloyds [13]. Our experiments suggest that the issuer is not vulnerable to this attack. Our tests with TSB [14] and Barclays [11] cards show the same result.

Listing A.4: Cryptogram confusion Attack Log (Failed)

```
Received response from card emulator:
00A404000E325041592E5359532E444446303100
SELECT 2PAY.SYS.DDF01
Sending 00A404000E325041592E5359532E444446303100
Received command from terminal emulator:
6F2B840E325041592E5359532E4444463031A519BF0C1661144F07A000000003
10109F0A080001050100000009000
Status code: 9000 Command successfully executed (OK).
  6F | len:2B    File Control Information (FCI) Template
      84 | len:14    DF Name: 325041592E5359532E4444463031
      A5 | len:19    Proprietary Information
        BF0C | len:16    File Control Information (FCI) Issuer
            Discretionary Data
              61 | len:14    Directory Entry
                  4F | len:7    Application Identifier (AID):
                      A0000000031010
                9F0A | len:8    Application Selection Registered
                    Proprietary Data list: 0001050100000000
Sending
6F2B840E325041592E5359532E4444463031A519BF0C1661144F07A000000003
10109F0A080001050100000009000
Received response from card emulator: 00A4040007A000000003101000
SELECT A0000000031010
Sending 00A4040007A000000003101000
Received command from terminal emulator:
6F578407A0000000031010A54C500A5649534120444542495487010 29F38189F
66049F02069F03069F1A0295055F2A029A039C019F37045F2D02656EBF0C1A9F
5A0531082608269F0A080001050100000000BF6304DF2001809000
Status code: 9000 Command successfully executed (OK).
```

```
  6F | len:57    File Control Information (FCI) Template
      84 | len:7    DF Name: A0000000031010
      A5 | len:4C    Proprietary Information
          50 | len:10    Application Label: 56495341204445424954
          87 | len:1    Application Priority Indicator: 02
        9F38 | len:18    Processing Options Data Object List (
            PDOL)
            9F66 | len:04   Terminal Transaction Qualifier (TTQ)
            9F02 | len:06   Amount, Authorised (Numeric)
            9F03 | len:06   Amount, Other (Numeric)
            9F1A | len:02   Terminal Country Code
              95 | len:05   Terminal Verification Results
            5F2A | len:02   Transaction Currency Code
              9A | len:03   Transaction Date
              9C | len:01   Transaction Type
            9F37 | len:04   Unpredictable Number
        5F2D | len:2    Language Preference: 656E
        BF0C | len:1A    File Control Information (FCI) Issuer
            Discretionary Data
            9F5A | len:5    Application Program Identifier:
                3108260826
            9F0A | len:8    Application Selection Registered
                Proprietary Data list: 0001050100000000
            BF63 | len:4    Unknown Payment System Tag: DF200180
Sending
6F578407A0000000031010A54C500A5649534120444542495487010 29F38189F
66049F02069F03069F1A0295055F2A029A039C019F37045F2D02656EBF0C1A9F
5A0531082608269F0A08000105010000000 0BF6304DF2001809000
Received response from card emulator:
80A8000023832136A040000000000001000000000000000826000000000000826
230505001F7FF7DF00
GPO command:
 9F66 | len  4    Terminal Transaction Qualifier (TTQ)
                  :36A04000
    EMV Mode supported (Byte 1 Bit 6)
    EMV contact chip supported (Byte 1 Bit 5)
    Online PIN supported (Byte 1 Bit 3)
    Signature supported (Byte 1 Bit 2)
    Online cryptogram required (Byte 2 Bit 8)
    Contact chip offline pin supported (Byte 2 Bit 6)
    Mobile device functionality supported (Byte 3 Bit 7)
 9F02 | len  6    Amount, Authorised (Numeric):
 000000000100
 9F03 | len  6    Amount, Other (Numeric): 000000000000
 9F1A | len  2    Terminal Country Code: 0826
```

```
   95 | len  5    Terminal Verification Results: 0000000000
 5F2A | len  2    Transaction Currency Code: 0826
   9A | len  3    Transaction Date: 230505
   9C | len  1    Transaction Type: 00
 9F37 | len  4    Unpredictable Number: 1F7FF7DF
Sending
80A8000023832136A040000000000001000000000000000826000000000000826
230505001F7FF7DF00
Received command from terminal emulator:  6984
Status code:  6984
Referenced data reversibly blocked (invalidated)
Sending 6984
```

A.5 OPay Participant Information Leaflet

A.5.1 Introduction

[Welcome, project title, investigators' names]

The main purpose is to study the usability of our proposed solution; OPay which is an orientation-based contactless payment solution against passive relay attacks. In this attack scenario, a malicious mPoS (mobile Point of Sale) terminal holder is able to steal money from people's bank cards by approaching their bags, pockets, etc (where it is more likely to put a bank card in) and make a contactless payment without anyone noticing. In our solution, we use the orientation data to make sure both the mPoS terminal and bank card are aligned with each other (having the same orientation data) before the payment is approved. It is based on the assumption that the attacker does not know the orientation of the card when it is placed in a bag or in a pocket, with limited chances of guessing. We also want to measure how the proposed solution has made the attack scenario difficult. Please note:

- The personal data about gender and age and the survey forms are collected and stored on a university-owned laptop, with only the access of the first investigator.
- The experimental data will be collected through a computer application.
- Collected data will not be identified and each participant will be given a Participant ID to relate different gathered data in the analyzing phase.

- These data will be processed in order to study the usability of the proposed solution.
- The two main researchers only will be able to access the data.
- No personal data will be transferred or shared to other organizations outside of the University, or outside of the EEA.

A.5.2 Experiments

Taking part in this study involves:

You will be asked to fill in a form that gathers information about your age range and your gender. You will also have the option not to provide any of these data. You will also be asked questions about the frequency of using contactless payment when you make a payment, and what devices you use specifically to make a contactless payment. These devices include cards, smartphones, smartwatches, etc. Filling in this form takes about 3 minutes.

Then, you will be asked to perform the following 4 series of experiments, each 5 times. To propose a solution against the proposed attack, we have designed two boards, one performs as a payment device (like a bank card), and the other one performs as the PoS terminal. If you have any difficulties in completing this part of the study, either picture of the experiment will be shown, or the investigator will show you how to perform each step of this experiment.

Experiment 1: You are required to hold the payment device board in an aligned way with the PoS terminal board in a way that they have the same orientation. Holding two devices near each other is equal to making a contactless transaction in our study. By holding two devices in an aligned way, we mean holding the payment device board in parallel with the PoS board. The PoS board is stable on the table, so you only need to hold the payment device board.

Experiment 2: In this experiment, you are asked to hold one of the boards (PoS terminal board) and move it randomly in 3D space. Any movement of your choice is appreciated, including rotating, flipping, etc. This experiment is designed to collect some orientation data and measure their randomness.

Experiment 3: In experiment 3, you are asked to perform as an attacker. We put the payment device board in a bag, and ask you to hold the PoS terminal board, approach the bag, and try to guess the board's orientation to get aligned with it. If you can guess the orientation of the payment device

board, it is equivalent to making a contactless transaction (which means the attacker is able to steal the money!). If the chance of guessing is low, it means our solution has been able to make this attack as hard as possible.

Experiment 4: This experiment is similar to experiment 3, except that we put the payment device board in a pocket. In this attack setup, we predict that the chances of guessing the orientation data are higher than in the previous experiment, as you (acting as the attacker) have more knowledge about the location and orientation of the payment device.

Finally, you will be asked to fill in 2 System Usability Scale (SUS) forms, one for the normal contactless payment, and one for the orientation-based contactless payment. Filling in these surveys takes about 5 minutes.

The experiments are supervised, in order to help you with your questions.

The whole study will not be recorded at any time.

No identifiable data are collected.

[Provided details regarding voluntary participation, the University of Warwick's data sharing policies, contact information for investigators for withdrawal from the study or further information, and acknowledgment section.]

A.6 User Study Survey Template

A.6.1 Introduction and Consent

Welcome, and thank you for participating in our research study. Our aim is to explore user perspectives on contactless payment, understanding its usage, perceived security risks, and protective measures. Before proceeding, please be aware of the following:

- This study is entirely voluntary. You may withdraw at any point by closing your browser.
- We collect minimal personal data, such as demographic information and Prolific IDs to manage participation and compensation and to ensure response quality.
- Your data will be securely stored, accessible only to our investigators for scientific analysis. The findings might be shared at conferences or in publications.
- The study has received approval from the University of Warwick research ethics committee.

Section 2 of this survey may make you think about potential fraud risks associated with contactless payment. If this causes you to worry, you can take the protective actions provided at the end of the survey to enhance the security of your contactless payments. For any questions or concerns, please contact mahshid.mehr-nezhad@warwick.ac.uk.]

A.6.2 General Knowledge and Preferences

[This section includes general questions about your contactless payment.]

1.1. How well do you know contactless payment? [I've Never heard of it, I've Heard of it but don't know what this is, I know what this is, but don't know how it works, I know generally how it works, I know very well how it works.]

1.2. In your own words, explain how contactless payment works. [long-answer text box]

1.3. Have you used contactless payment in the past six months? [Yes, No, I don't remember]

1.4. How often do you use contactless payment? [Several times a day, One or two times a day, One or two times a week, One or two times a month, Never]

1.5. Which of the following contactless payment devices do you use? (Select all that apply) [Contactless credit or debit card, Mobile contactless payment (e.g. Apple Pay, Google Pay), Wearable contactless payment (e.g. Smartwatch, Smart jewelry, accessories, etc.), other]

1.6. What do you like about contactless technology? (Select all that apply)

[It is fast, It is convenient, It is secure, other]

1.7. What do you dislike about contactless technology? (Select all that apply)

[Technical issues (e.g. connectivity problems, device compatibility), Lack of familiarity with the technology, Concerns about security or fraud, Its maximum payment cap, other]

1.8. How often have you used the "Express Transit" mode in a contactless transaction to buy tickets on Transport For London (TFL)? (when you do not need to wake or unlock your device or authenticate with Face ID, Touch ID, or your passcode)?

[Never, Almost Never, Occasionally, Frequently, Always]

1.9. If you have used the "Express Transit" mode, please share your experience with it and any opinions and concerns you may have.

[Long-answer text paragraph]

1.10. How important is it to you that businesses (shops, cafes, etc.) offer contactless payment options?

[Not important, Somewhat important, Very important]

1.11. Have you ever used a contactless method to withdraw cash from an ATM?

[Yes, No]

1.12. Would you be interested in using a contactless card or digital wallet to withdraw cash from an ATM in the future?

[Yes, I am interested, No, I prefer the traditional card and PIN method to withdraw cash, I am unsure/neutral]

A.6.3 Perception on Contactless Payment Security

Description: Contactless technology allows users to make payments by simply tapping their contactless-enabled devices, like cards, smartphones, or wearable, close to a

contactless reader. It uses Near Field Communication (NFC) technology, which enables the transfer of data over short distances through the air. NFC has a range of a few centimeters, ensuring that payment information is only transmitted to the intended recipient. During a contactless payment, the payment device sends transaction data to the nearby terminal using NFC. The terminal then forwards the information to the bank for authorization. The bank reviews and approves or rejects the payment, notifying the terminal of the decision. This section focuses on attacks targeting contactless payment methods. Please read each attack description carefully before answering the related questions.

2.1. How concerned are you about the general security and privacy of your contactless payments?

[Very Concerned, Concerned, Neutral, Not Concerned, Not At All Concerned]

2.2. How secure do you think each contactless payment method is?

[Table with a list of payment devices in the rows Contactless credit/debit card, Mobile Devices(e.g. Apple Pay, Google Pay), Wearable Devices(e.g. Smartwatch) and familiarity level in the columns (Very Secure, Secure, Neutral, Not Secure, Not At All Secure)

2.3 to 2.14: asks users to what extent they think each of the following six attacks is feasible (Feasible, Somewhat Feasible, Not Feasible) and how concerned they are about each attack in particular (Very Concerned, Concerned, Neutral, Not Concerned, Not At All Concerned)

A) Data Leakage Attack: Attackers utilize different techniques to gain unauthorized access to sensitive information from contactless payment cards. This includes extracting crucial data such as the Primary Account Number and expiry date, which subsequently compromises the overall security and confidentiality of the cardholders' personal and financial data.

[Example included from Table 6.2.]

B) Relay Attack: Although the typical NFC range is limited to a few centimeters, it is possible to extend this range significantly. Attackers can intercept and relay payment information between a payment card and a distant terminal, utilizing two devices. The initial device captures payment data and transmits it to the second device, which then relays it to the payment terminal, allowing attackers to make unauthorized transactions in real-time without users' knowledge.

[Example included from Table 6.2.]

C) Pre-play Attack: Attackers record payment information during a legitimate transaction, preserving it for future fraudulent activities. By compromising the payment terminal, they intercept and store the payment data without the user's awareness, enabling them to conduct fraudulent transactions at a later time. This attack is difficult to detect during a legitimate transaction, as the payment terminal appears to function normally.

[Example included from Table 6.2.]

D) Counterfeit Card Replica Attack: Attackers intercept and extract all the magnetic stripe data from a physical payment card, like a credit or debit card, either through the NFC interface or by swiping the card, and subsequently store the information for later use in creating a replicate by writing it onto a blank magnetic stripe card that can be used later for fraudulent activities.

[Example included from Table 6.2.]

E) Contactless Payment Limit Bypass Attack: Attackers exploit the contactless transaction limit to steal money by surpassing the set limit. In the UK, where the current limit is £100 for a single contactless payment without a PIN, attackers initiate a payment to the victim's payment device. Using specialized equipment, typically smartphones, they manipulate the payment information during the transaction and authorize high-value contactless transactions without requiring a PIN.

[Example included from Table 6.2.]

F) Lock-screen Bypass Attack: Attackers exploit the "transit mode" feature in lock-screen payment devices (e.g., smartphones), which is designed for convenient fare payment on public transport without requiring unlocking the phone. By bypassing the lock-screen, they carry out unauthorized transactions without the victim's knowledge. This can occur through close proximity, specialized equipment, or terminal compromise.

[Example included from Table 6.2.]

2.15. How concerned are you about the general security and privacy of your contactless payments?

[Very Concerned, Concerned, Neutral, Not Concerned, Not At All Concerned]

A.6.4 Protective Actions

[Description: This section covers protective actions for contactless payments.]

3.1. How do you monitor your payment activities and account(s)?

[Online banking on PC or laptop, Mobile banking app, keeping and checking purchase receipts, Someone else does it for me (parent, partner, lawyer, etc.), Other]

3.2. What measures do you take for your contactless payment security and privacy?

- I use RFID-blocking wallets or card sleeves,
- I am cautious when using contactless payment in unfamiliar or untrusted environments,
- I disable the "express transit" feature on my smartphone/smartwatch if not using it,
- I switch off the NFC sensor on my mobile phone,
- I limit the maximum I can spend (spread over multiple payments) before I need to enter my PIN in online or mobile banking,

- I limit the maximum transaction amount for contactless payment to a single tap in online or mobile banking,
- I do not carry cards or add cards to my digital wallets with lots of funds,
- I'd ask the bank if they can issue a card without a contactless feature,
- I switch off contactless transactions entirely through settings in online or mobile banking,
- I enable passive acknowledging of transactions (such as SMS notifications, bank calls, etc.),
- I check my bank receipts and bank accounts regularly,
- I ask to receive a monthly paper statement and review it in detail,
- Nothing,
- Other: ———-

3.3. What are other actions that you think are effective in protecting your contactless payment security and privacy?

[Long-answer text paragraph]

A.6.5 Demographic Data, Feedback, and Compensation

Questions about age, gender, the highest level of education, and users' feedback is asked, and a code for compensation is provided to be claimed on Prolific [108].

Bibliography

[1] Adafruit. Adafruit bluefruit ble sniffer. Available at https://www.adafruit.com/product/2269. Accessed 10 May 2022.

[2] Mohammed A Al-Sharafi, Noor Al-Qaysi, Noorminshah A Iahad, and Mostafa Al-Emran. Evaluating the sustainable use of mobile payment contactless technologies within and beyond the covid-19 pandemic using a hybrid sem-ann approach. *International Journal of Bank Marketing*, 40(5):1071–1095, 2022.

[3] Lukas Aldag, Karen Renaud, Benjamin Berens, Reyhan Duezguen, Mattia Mossano, and Melanie Volkamer. Reporting on insights gained into uk citizens' perceptions of contactless card risks. *Karlsruher Institute for technology (KIT). doi*, 10, 2020.

[4] Mohammed Aamir Ali, Budi Arief, Martin Emms, and Aad van Moorsel. Does the online card payment landscape unwittingly facilitate fraud? *IEEE Security & Privacy*, 15(2):78–86, 2017.

[5] Ross Anderson. *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons, 2020.

[6] Apple. Applepay digital wallet. Available at https://www.apple.com/uk/apple-pay/, . Accessed 11 January 2023.

[7] Apple. Applepay express transit mode. Available at https://www.apple.com/uk/apple-pay/transport/, . Accessed 11 January 2023.

[8] Apple. Where you can travel on public transport using apple pay. Available at https://support.apple.com/en-gb/HT207958, . Accessed 1 August 2023.

[9] Apple. Apple watch series 8. Available at https://www.apple.com/uk/shop/buy-watch/apple-watch, . Accessed 29 March 2023.

[10] Ayden. Tap to pay on iphone. https://www.adyen.com/devices/tap-to-pay-on-iphone. Accessed: 20 July 2023.

[11] Barclays Bank. Barclays bank personal banking. https://www.barclays.co.uk/, . Accessed: 30 July 2023.

[12] Citi Bank. Merchant category codes. Available at https://www.citibank.com/tts/solutions/commercial-cards/assets/docs/govt/Merchant-Category-Codes.pdf, 2015. Accessed 11 January 2023.

[13] Lloyds Bank. Lloyds bank ready-made investments. https://www.lloydsbank.com/, . Accessed: 30 July 2023.

[14] TSB Bank. Tsb bank: Personal banking. https://www.tsb.co.uk/personal/, . Accessed: 30 July 2023.

[15] Barclays. Barclays contactless cash. Available at https://www.barclays.co.uk/ways-to-bank/contactless-cash/. Accessed 11 January 2023.

[16] David Basin, Ralf Sasse, and Jorge Toro-Pozo. The emv standard: Break, fix, verify. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1766–1781. IEEE, 2021.

[17] David Basin, Ralf Sasse, and Jorge Toro-Pozo. Card brand mixup attack: Bypassing the PIN in non-visa cards by using them for visa transactions. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 179–194. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/basin.

[18] David Basin, Patrick Schaller, and Jorge Toro-Pozo. Inducing authentication failures to bypass credit card pins. In *32rd USENIX Security Symposium (USENIX Security*, 2023.

[19] Stefano Berlato and Mariano Ceccato. A large-scale study on the adoption of anti-debugging and anti-tampering protections in android apps. *Journal of Information Security and Applications*, 52:102463, 2020.

[20] Thomas Bocek, Christian Killer, Christos Tsiaras, and Burkhard Stiller. An nfc relay attack with off-the-shelf hardware and software. In Rémi Badonnel, Robert Koch, Aiko Pras, Martin Drašar, and Burkhard Stiller, editors, *Management and Security in the Age of Hyperconnectivity*, pages 71–83, Cham, 2016. Springer International Publishing. ISBN 978-3-319-39814-3.

[21] John Brooke. Sus: a "quick and dirty'usability. *Usability evaluation in industry*, 189, 1996.

[22] Karoline Busse, Mohammad Tahaei, Katharina Krombholz, Emanuel von Zezschwitz, Matthew Smith, Jing Tian, and Wenyuan Xu. Cash, cards or cryptocurrencies? a study of payment culture in four countries. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 200–209. IEEE, 2020.

[23] Android Central. No, the oneplus 8 pro doesn't have an x-ray camera — here's what's actually happening. Available at `https://www.androidcentral.com/no-oneplus-8-pro-doesnt-have-x-ray-camera`. Accessed 15 June 2021.

[24] Tom Chothia, Flavio D Garcia, Joeri De Ruiter, Jordi Van Den Breekel, and Matthew Thompson. Relay cost bounding for contactless emv payments. In *Financial Cryptography and Data Security: 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers 19*, pages 189–206. Springer, 2015.

[25] Alexei Czeskis, Karl Koscher, Joshua R Smith, and Tadayoshi Kohno. Rfids and secret handshakes: Defending against ghost-and-leech attacks and unauthorized reads with context-aware communications. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 479–490, 2008.

[26] Joeri De Ruiter and Erik Poll. Formal analysis of the emv protocol suite. In *Joint Workshop on Theory of Security and Applications*, pages 113–129. Springer, 2011.

[27] Java Decompiler. Java online decompiler. Available at `http://www.javadecompilers.com/apk`. Accessed 13 May 2022.

[28] James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006.

[29] Saar Drimer, Steven J Murdoch, et al. Keep your enemies close: Distance bounding against smartcard relay attacks. In *USENIX security symposium*, volume 312, 2007.

[30] Martin Emms, Budi Arief, Troy Defty, Joseph Hannon, Feng Hao, et al. The dangers of verify pin on contactless cards. *School of Computing Science Technical Report Series*, 2012.

[31] Martin Emms, Budi Arief, Nicholas Little, and Aad van Moorsel. Risks of offline verify pin on contactless cards. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, pages 313–321, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-39884-1.

[32] Martin Emms, Budi Arief, Leo Freitas, Joseph Hannon, and Aad van Moorsel. Harvesting high value foreign currency transactions from emv contactless credit cards without the pin. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 716–726, 2014.

[33] Martin Emms et al. Practical attack on contactless payment cards. *HCI 2011: Health Wealth and Happiness*, 2011.

[34] EMVCo. Emv 3-d secure. Available at `https://www.emvco.com/emv-technologies/3-d-secure/`, . Accessed 9 January 2023.

[35] EMVCo. Emv contact chip. Available at `https://www.emvco.com/emv-technologies/emv-contact-chip/`, . Accessed 9 January 2023.

[36] EMVCo. Emv contactless chip. Available at `https://www.emvco.com/emv-technologies/emv-contactless-chip/`, . Accessed 9 January 2023.

[37] EMVCo. Emvco. Available at `https://www.emvco.com/`, . Accessed.

[38] EMVCo. Emv mobile. Available at `https://www.emvco.com/emv-technologies/mobile/`, . Accessed 9 January 2023.

[39] EMVCo. Emv payment tokenisation. Available at `https://www.emvco.com/emv-technologies/payment-tokenisation/`, . Accessed 9 January 2023.

[40] EMVCo. Emv qr code. Available at `https://www.emvco.com/emv-technologies/qr-codes/`, . Accessed 9 January 2023.

[41] EMVCo. Emv secure remote commerce (src). Available at `https://www.emvco.com/emv-technologies/secure-remote-commerce/`, . Accessed 9 January 2023.

[42] EMVCo. *EMV® Contactless Specifications for Payment Systems Book, Book C-3, Kernel 3 Specification* . February 2016. Version 2.6.

[43] EMVCo. *EMV® Integrated Circuit Card Specifications for Payment Systems: Book 2 Security and Key Management.* 2022. Version 4.4.

[44] EMVCo. *EMV® Contactless Specifications for Payment Systems, Book C-8. Kernel 8 Specification.* October 2022. Version 1.0.

[45] EMVCo. *EMV® Contactless Specifications for Payment Systems Book, Book C-2, Kernel 2 Specification* . March 2022. Version 2.10.

[46] EMVCo. *EMV® Contactless Specifications for Payment Systems: Book A - Architecture and General Requirements.* June 2023. Version 2.11.

[47] EMVCo. *EMV® Contactless Specifications for Payment Systems: Book B, Entry Point Specification.* June 2023. Version 2.11.

[48] EMVCo. *EMV® Contactless Specifications for Payment Systems Book, Book C-2, Kernel 2 Specification* . June 2023. Version 2.11.

[49] EMVCo. *EMV® Contactless Specifications for Payment Systems Book, Book C-3, Kernel 3 Specification* . June 2023. Version 2.11.

[50] EMVCo®. 4 key features of the new emv® contactless kernel specification. Available at https://www.emvco.com/knowledge-hub/4-key-features-of-the-new-emv-contactless-kernel-specification/. Accessed 30 May 2022.

[51] American Express. American express: Credit cards, rewards, travel, and business services. https://www.americanexpress.com/en-gb/, 2023. Accessed: 15 May 2023.

[52] Peter Fillmore. Crash and pay: Owning and cloning payment devices. *BlackHat*, 2015.

[53] UK Finance. Uk payment markets summary 2022. Available at https://www.ukfinance.org.uk/system/files/2022-08/UKF%20Payment%20Markets%20Summary%202022.pdf, . Accessed 11 January 2023.

[54] United Kingdom Finance. The problems with contactless cards. Available at http://www.contactlesspaymentcards.com/problems-with-contactless-cards.php, . Accessed 8 September 2021.

[55] Fitbit. Make hands-free purchases easily with fitbit pay. Available at https://www.fitbit.com/global/us/technology/fitbit-pay. Accessed 29 March 2023.

[56] Forbes. What is pos and how does it work? Available at https://www.forbes.com/advisor/in/banking/what-is-pos-and-how-does-it-work/, 2023. Accessed 11 January 2023.

[57] Lishoy Francis, Gerhard Hancke, Keith Mayes, and Konstantinos Markantonakis. Practical relay attack on contactless transactions by using nfc mobile phones. *IACR Cryptology ePrint Archive*, 2011:618, 01 2011. doi: 10.3233/978-1-61499-143-4-21.

[58] WesLee Frisby, Benjamin Moench, Benjamin Recht, and Thomas Ristenpart. Security analysis of smartphone point-of-sale systems. In *WOOT*, pages 22–33, 2012.

[59] Leigh-Anne Galloway. It only takes a minute to clone a credit card, thanks to a 50-year-old problem. *Tech Report*, 2020.

[60] Leigh-Anne Galloway and Tim Yunusov. For the love of money: Finding and exploiting vulnerabilities in mobile point of sales systems. Available at `https://leigh-annegalloway.com/for-the-love-of-money/`. Accessed 11 January 2023.

[61] Leigh-Anne Galloway and Tim Yunusov. First contact: New vulnerabilities in contactless payments. *Black Hat Europe*, 2019, 2019.

[62] Aznida Wati Abdul Ghani, Abdul Hafaz Ngah, and Azizul Yadi Yaakop. Why should i continue using it? factors influencing continuance intention to use e-wallet: The sor framework. In *International Conference on Information Systems and Intelligent Applications: ICISIA 2022*, pages 1–16. Springer, 2022.

[63] Google. Googlepay screen lock. Available at `https://support.google.com/wallet/answer/12059519?co=GENIE.Platform=Android&hl=en&visit_id=638043049090351002-1694550852&rd=1`, . Accessed 17 November 2022.

[64] Google. Googlepay digital wallet. Available at `https://pay.google.com/intl/en_uk/about/`, . Accessed 11 January 2023.

[65] United Kingdom Government. 2021 budget plan. Available at `https://www.gov.uk/government/publications/budget-2021-documents`. Accessed 01 June 2021.

[66] Iakovos Gurulian, Raja Naeem Akram, Konstantinos Markantonakis, and Keith Mayes. Preventing relay attacks in mobile transactions using infrared light. In *Proceedings of the Symposium on Applied Computing*, pages 1724–1731, 2017.

[67] Iakovos Gurulian, Gerhard P Hancke, Konstantinos Markantonakis, and Raja Naeem Akram. May the force be with you: Force-based relay attack detection. In *International Conference on Smart Card Research and Advanced Applications*, pages 142–159. Springer, 2017.

[68] Iakovos Gurulian, Konstantinos Markantonakis, Eibe Frank, and Raja Naeem Akram. Good vibrations: artificial ambience-based relay attack detection. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 481–489. IEEE, 2018.

[69] Tzipora Halevi, Di Ma, Nitesh Saxena, and Tuo Xiang. Secure proximity detection for nfc devices based on ambient sensor data. In *European Symposium on Research in Computer Security*, pages 379–396. Springer, 2012.

[70] Jian Yuan Haoqi Shan. Man in the nfc. *DEF CON 25*, 2017.

[71] Thomas S. Heydt-Benjamin, Daniel V. Bailey, Kevin Fu, Ari Juels, and Tom O'Hare. Vulnerabilities in first-generation rfid-enabled credit cards. In Sven Dietrich and Rachna Dhamija, editors, *Financial Cryptography and Data Security*, pages 2–14, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-77366-5.

[72] Muhamad Fitri Ismail, Mohd Akmal Rohiat, Azlan Salim, and Dewi Eka Murniati. Customer experience towards contactless payment service practices in the pandemic covid-19 era. a case study: Fast food restaurants. *Journal of Technology and Humanities*, 3(1):1–6, 2022.

[73] ISO. Iso 8583: Financial transaction card originated messages, 2003. Standard.

[74] ISO. 14443-1: 2018 – cards and security devices for personal identification – contactless proximity objects – part 1: Physical characteristics, 2018. Standard.

[75] ISO. 14443-3: 2018 – identification cards – contactless integrated circuit cards – proximity cards – part 3: Initialization and anticollision, 2018. Standard.

[76] ISO. 14443-2: 2020 – cards and security devices for personal identification – contactless proximity objects – part 2: Radio frequency power and signal interface, 2020. Standard.

[77] iZettle. In-app pairing guide. Available at `https://developer.zettle.com/docs/ios-sdk/user-guides/manage-in-app-pairing`. Accessed 12 March 2023.

[78] iZettle. izettle card reader. Available at `https://www.izettle.com/`, 2023. Accessed 11 January 2023.

[79] Rong Jin, Liu Shi, Kai Zeng, Amit Pande, and Prasant Mohapatra. Magpairing: Pairing smartphones in close proximity using magnetometers. *IEEE Transactions on Information Forensics and Security*, 11(6):1306–1320, 2015.

[80] Ricardo J. Rodriguez Jose Vila. Relay attacks in emv contactless cards with android ots devices. *HITBSecConf*, 2015.

[81] Kerv. Safe and easy payment, k ring: Powered by mastercard. Available at `https://mykring.com/en/bank/abn-amro/`. Accessed 29 March 2023.

[82] Alfred Kobsa, Rahim Sonawalla, Gene Tsudik, Ersin Uzun, and Yang Wang. Serial hook-ups: a comparative usability study of secure device pairing methods. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12, 2009.

[83] M Kok, JD Hol, and TB Sch"on. Using inertial sensors for position and orientation estimation. *Foundations and Trends in Signal Processing*, 11:1–153, 2017.

[84] MWR Labs. Mission mpossible: Mobile card payment security. Available at https://www.youtube.com/watch?v=iwOP1hoVJEE. Accessed 11 January 2023.

[85] Eddie Lee. Nfc hacking: The easy way. In *Defcon hacking conference*, volume 20, pages 63–74, 2012.

[86] Makayla Lewis and Mark Perry. Follow the money: Managing personal finance digitally. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.

[87] Di Ma, Nitesh Saxena, Tuo Xiang, and Yan Zhu. Location-aware and safer cards: enhancing rfid security and privacy via location sensing. *IEEE transactions on dependable and secure computing*, 10(2):57–69, 2012.

[88] Mastercard. Ways to pay, pay at pump. Available at https://www.mastercard.co.uk/en-gb/personal/ways-to-pay/pay-at-pump.html, . Accessed 11 January 2023.

[89] Mastercard. Mastercard tap to pay on iphone. Available at https://partner.visa.com/site/programs/visa-ready/tap-to-phone.html, . Accessed 11 January 2023.

[90] Mastercard. Tap to pay on iphone. https://www.mastercard.com/global/en/business/overview/start-accepting/mobile-pos/iphone.html, . Accessed: 20 July 2023.

[91] Mastercard. Swiping left on magnetic stripes. https://www.mastercard.com/news/perspectives/2021/magnetic-stripe/, . [Accessed 26 May 2023].

[92] Mastercard. Experience the world with mastercard. https://www.mastercard.co.uk/en-gb.html, 2023. Accessed: 15 May 2023.

[93] McLear. Ringpay. Available at https://mclear.com/product/payment-ring/. Accessed 29 March 2023.

[94] Mahshid Mehr Nezhad and Feng Hao. Opay: an orientation-based contactless payment solution against passive attacks. In *Annual Computer Security Applications Conference*, pages 375–384, 2021.

[95] Maryam Mehrnezhad, Feng Hao, and Siamak F Shahandashti. Tap-tap and pay (ttp): Preventing the mafia attack in nfc payment. In *International Conference on Research in Security Standardisation*, pages 21–39. Springer, 2015.

[96] Maryam Mehrnezhad, Mohammed Aamir Ali, Feng Hao, and Aad van Moorsel. Nfc payment spy: A privacy attack on contactless payments. In Lidong Chen, David McGrew, and Chris Mitchell, editors, *Security Standardisation Research*, Cham, 2016. Springer International Publishing. ISBN 978-3-319-49100-4.

[97] Alexandrea Mellen, John Moore, and Artem Losev. Mobile point of scam: Attacking the square reader.

[98] Julien MILLAU. Pro credit card reader nfc. `https://play.google.com/store/apps/details?id=com.github.devnied.emvnfccard.pro&hl=en&gl=US`. Google Play Store.

[99] Mitmproxy. How mitmproxy works. Available at `https://docs.mitmproxy.org/stable/concepts-howmitmproxyworks/`. Accessed 11 January 2023.

[100] Uma Thevi Munikrishnan, Abdullah Al Mamun, Nicole Kok Sue Xin, Ham Siu Chian, and Farzana Naznen. Modelling the intention and adoption of cashless payment methods among the young adults in malaysia. *Journal of Science and Technology Policy Management*, 2022.

[101] Bank of England. How do card payments work? Available at `https://www.bankofengland.co.uk/explainers/how-do-card-payments-work`. Accessed 20 July 2023.

[102] Kristin Paget. Credit card fraud: The contactless generation. *ShmooCon*, 2012.

[103] Patrickfav. Apk tool- a tool for reverse engineering android apk files. Available at `https://ibotpeaches.github.io/Apktool/`, . Accessed 13 May 2022.

[104] Patrickfav. Uber apk signer. Available at `https://github.com/patrickfav/uber-apk-signer`, . Accessed 13 May 2022.

[105] Google Pay. Pay with a qr code (singapore only). `https://support.google.com/googlepay/answer/9984037?hl=en`. Accessed: 20 July 2023.

[106] PayPal. The world is now your marketplace. `https://www.paypal.com/in/webapps/mpp/ecommerce`. Accessed: 20 July 2023.

[107] Gary Pritchard, John Vines, and Patrick Olivier. Your money's no good here: The elimination of cash payment on london buses. In *Proceedings of the 33rd*

Annual ACM Conference on Human Factors in Computing Systems, pages 907–916, 2015.

[108] Prolific. Prolific, quickly find research participants you can trust. https://www.prolific.com, 2023. Accessed: 15 May 2023.

[109] Andreea-Ina Radu, Tom Chothia, Christopher J.P. Newton, Ioana Boureanu, and Liqun Chen. Practical emv relay protection. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1737–1756, 2022. doi: 10.1109/SP46214.2022.9833642.

[110] Michael Roland and Josef Langer. Cloning credit cards: A combined pre-play and downgrade attack on EMV contactless. In *7th USENIX Workshop on Offensive Technologies (WOOT 13)*, Washington, D.C., August 2013. USENIX Association. URL https://www.usenix.org/conference/woot13/workshop-program/presentation/roland.

[111] Mike Ryan. Crackle. Available at https://github.com/mikeryan/crackle. Accessed 24 May 2022.

[112] Mike Ryan. Bluetooth: With low energy comes low security. In *7th {USENIX} Workshop on Offensive Technologies ({WOOT} 13)*, 2013.

[113] Samsung. Galaxy watches. Available at https://www.samsung.com/uk/watches/galaxy-watch/, . Accessed 29 March 2023.

[114] Samsung. Samsungpay digital wallet. Available at https://www.samsung.com/uk/samsung-pay/, . Accessed 11 January 2023.

[115] Aleksandr Shevelev. Nfc reader. https://play.google.com/store/apps/details?id=info.iso8583.nfcreader&hl=en&gl=US&pli=1. Google Play Store.

[116] Wesam Shishah and Soha Alhelaly. User experience of utilising contactless payment technology in saudi arabia during the covid-19 pandemic. *Journal of Decision Systems*, 30(2-3):282–299, 2021.

[117] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N Asokan. Drone to the rescue: Relay-resilient authentication using ambient multi-sensing. In *International Conference on Financial Cryptography and Data Security*, pages 349–364. Springer, 2014.

[118] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N Asokan. Sensor-based proximity detection in the face of active adversaries. *IEEE Transactions on Mobile Computing*, 18(2):444–457, 2018.

[119] shroudedcode. apk-mitm. Available at https://github.com/shroudedcode/apk-mitm. Accessed 13 May 2022.

[120] Bluetooth SIG. Bluetooth core specification, v5.2. Available at https://www.bluetooth.com/specifications/specs/core-specification-5-2/. Accessed 9 May 2022.

[121] Smasung. Set up your samsung pay transit card. Available at https://www.samsung.com/au/support/mobile-devices/samsung-pay-transit-card-setup/, . Accessed 11 January 2023.

[122] Smasung. Frequently asked questions. Available at https://www.samsung.com/uk/samsung-pay/faq/, . Accessed 1 August 2023.

[123] Luigi Sportiello and Andrea Ciardulli. Long distance relay attack. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, pages 69–85. Springer, 2013.

[124] Square. Square card reader. Available at https://squareup.com/gb/en, . Accessed 11 January 2023.

[125] Square. Accept and record payments on your computer. Available at https://squareup.com/gb/en/payments/virtual-terminal, . Accessed 11 January 2023.

[126] Square. Take contactless payments with just your iphone. https://squareup.com/us/en/payments/tap-to-pay, . Accessed: 20 July 2023.

[127] Statista. Market share of global general purpose card brands american express, diners/discover, jcb, mastercard, unionpay and visa from 2014 to 2022, based on number of transactions. https://www.statista.com/statistics/278970/share-of-purchase-transactions-on-global-credit-cards/.

[128] Aleksei Stennikov. Nfc mitm. https://github.com/a66at/NFCMiTM. Accessed 6 March 2023.

[129] Douglas R Stinson. *Cryptography: theory and practice*. Chapman and Hall/CRC, 2005.

[130] Stripe. Tap to pay. https://stripe.com/docs/terminal/payments/setup-reader/tap-to-pay, . Accessed: 20 July 2023.

[131] Stripe. A complete payments platform for e-commerce. https://stripe.com/gb/use-cases/ecommerce, . Accessed: 20 July 2023.

[132] Ahren Studer, Timothy Passaro, and Lujo Bauer. Don't bump, shake on it: The exploitation of a popular accelerometer-based smart phone exchange and its secure replacement. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 333–342, 2011.

[133] Sumup. Sumup card reader. Available at `https://www.sumup.com/en-gb/`. Accessed 11 January 2023.

[134] Miura Systems. Miura card reader. Available at `https://www.miurasystems.com/`. Accessed 11 January 2023.

[135] Aleksei Stennikov Timur Yunusov, Artem Ivachev. New vulnerabilities in public transport schemes for apple pay, samsung pay, gpay. *White Paper*, 2021.

[136] Hien Thi Thu Truong, Xiang Gao, Babins Shrestha, Nitesh Saxena, N Asokan, and Petteri Nurmi. Comparing and fusing different sensor modalities for relay attack resistance in zero-interaction authentication. In *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 163–171. IEEE, 2014.

[137] Jordi van den Breekel. Relaying emv contactless transactions using off-the-shelf android devices. *BlackHat Asia, Singapore*, 2015.

[138] Aditya Vashistha, Richard Anderson, and Shrirang Mare. Examining the use and non-use of mobile payment systems for merchant payments in india. In *Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–12, 2019.

[139] Vidushi Vatsa and Bhawna Agarwal. Factors impacting adoption and continuous use of contactless digital payments in the new normal. *International Journal of Electronic Finance*, 11(4):317–344, 2022.

[140] Visa. Pay at pump, self-service petrol payments with visa. Available at `https://www.visa.co.uk/pay-with-visa/pay-at-pump.html`, . Accessed 11 January 2023.

[141] Visa. Visa tap to phone. Available at `https://partner.visa.com/site/programs/visa-ready/tap-to-phone.html`, . Accessed 11 January 2023.

[142] Visa. Visa, a trusted leader in digital payments. `https://www.visa.co.uk/`, 2023. Accessed: 15 May 2023.

[143] Dominik Wermke, Nicolas Huaman, Yasemin Acar, Bradley Reaves, Patrick Traynor, and Sascha Fahl. A large scale investigation of obfuscation use in google play. pages 222–235, 2018.

[144] Timur Yunusov. Hand in your pocket without you noticing: Current state of mobile wallet security. *Black Hat Europe*, 2021.

[145] Timur Yunusov. How to clone google pay/mastercard transactions? Available at `https://www.paymentvillage.org/blog/how-to-clone-google-paymastercard-transactions`, 2022. 20 March 2023.

[146] Timur Yunusov. Modern emv and nfc cardholder verification issues the cryptogram confusion attack. Available at `https://www.paymentvillage.org/blog/modern-emv-and-nfc-cardholder-verification-issues`, 2022. 20 March 2023.

[147] Qingyu Zhang, Salman Khan, Mei Cao, and Safeer Ullah Khan. Factors determining consumer acceptance of nfc mobile payment: An extended mobile technology acceptance model. *Sustainability*, 15(4):3664, 2023.