

Apple Pay: How different is it from other Pay solutions, what role does tokenisation play, and to what degree can Card not Present payment benefit from Apple Pay in future

Marcel Fehr

Technical Report

RHUL-ISG-2018-3

3 April 2018



Information Security Group
Royal Holloway University of London
Egham, Surrey, TW20 0EX
United Kingdom

MARCEL FEHR

Student Number: 130263579

Apple Pay: How different is it from other 'Pay' solutions, what role does tokenisation play, and to what degree can Card not Present payment benefit from Apple Pay in future



Royal Holloway University of London

Information Security Group

Egham, Surrey, TW20 0EX

United Kingdom

Supervisor: Professor Kostas Markantonakis

Submitted as part of the requirements for the award of the MSc in Information Security at Royal Holloway, University of London.

I declare that this assignment is all my own work and that I have acknowledged all quotations from published or unpublished work of other people. I also declare that I have read the statements on plagiarism in Section 1 of the Regulations Governing Examination and Assessment Offences, and in accordance with these regulations I submit this project report as my own work.

Signature:

Date:

Acknowledgements

Thanks to my partner Doris and her incredible patience, the University of London and all the people involved providing the lectures and those working in the background make the distance learning programme happen. It was and still is a great experience to be part of the distance learning program.

Abstract

We are living in a world where smartphones follow us at every turn. We are used to 'bringing our own device'. Why not introduce secure mobile payments as part of our daily purchasing experience? Indeed, the trend in consumer preference for mobile wallets over physical wallets is well documented, and several recent surveys [4] indicate a mild surge in the use of mobile wallets. This surge is mainly caused by the publicity around the launch of the three 'Pay' solutions, namely Apple Pay, Samsung Pay, and Android Pay. At the same time, questions emerge about using a smartphone as a payment device. People wonder whether it is safe enough or if the sensitive cardholder information is sufficiently protected. Moreover, privacy concerns surface concerning the extensive collection of metadata.

This paper focuses on the security aspects of Apple Pay as a mobile payment solution at a point of sales (PoS) and its application of EMV tokenization and compares it to Samsung Pay and Android Pay. The aim is to provide insight into the different payment ecosystems, allow the reader to understand the deviating approaches to wallet security, and provide an understanding of what can go wrong and what Apple Pay does well concerning security.

The three wallets implement the mobile payment solution differently, but all are compatible to the EMV contactless payment specification and tokenization and are supported by the card schemes. They allow using a smartphone as a contactless credit card (near field communication device) at a PoS and are compatible to MasterCard's *PayPass* and Visa's *payWave* specification.

The project demonstrates that following the EMV tokenization specification greatly improves the security of contactless payment at a PoS irrespective of the solution. This is because the use of tokenization combined with dynamic EMV payment cryptograms *renders captured payment details mostly useless for cross-channel fraud in card not present transactions*. Additionally, using the smartphone as a payment device and its ability to provide additional metadata, facilitates enhanced fraud analytics. This comes along with robust mechanisms for cardholder identification and device authentication, applying fingerprints and one-time passwords to name a few. The new 3D-Secure 2.0 specification [80] will play a significant role in providing metadata.

The threat and vulnerability analysis of the Apple Pay ecosystem has not revealed any weaknesses but outlines that the increased number of stakeholders (e.g. TSP, wallet provider) widened the possible attack surface. The card enrolment process is an attractive target for fraudsters and must be watched to prevent enrolment of stolen credit cards. Another important aspect is the security posture of the payment device. Both, Samsung Pay with its trusted execution environment (TEE) and Apple Pay with the secure element technology follow the Security by Design approach. Android provides security with a multi-layered approach. It uses Host Card Emulation (HCE), where tokenization is employed and the limited use keys are replenished in time through a cloud connection.

Besides PoS security improvements for contactless payments, the secure remote payment path will soon experience important changes as it is expected that the Card Present fraud figures will further drop in favour of a significant rise in CNP figures. This constellation has been analysed by the Boston Reserve Bank [61]. In the UK, fraud statistics [56] show a significant 20 % increase in CNP e-commerce frauds to the year before. This is where Apple Pay's remote secure payment implementation can play an important role in the future. The option to widen the scope from mobile in-app purchase using an EMV payment token and cryptograms to third party devices sounds promising. *This facilitates the EMV cryptographic strength to the CNP environment* and would help to minimize fraud. From my point of view, the introduction of EMV at PoS in the United States could have been a strategic step to prepare the United States' outdated payment infrastructure for the new mobile payment environment, including secure remote payment.

Overall, mobile payment solutions have a lot to offer regarding providing metadata for advanced fraud analytics and prevention, strong cardholder identification or the small effort it needs to manage the tokenized credit cards compared to the physical replacement tasks due to fraud, loss or theft. All three 'Pay' solutions will have their share in the mobile payment market.

Keywords

Apple Pay, Android Pay, Samsung Pay, tokenisation, mobile payment, host card emulation, HCE

Contents

Acknowledgements	i
Abstract	ii
Keywords	iii
List of Figures	vi
List of Tables	VIII
Chapter 1 Introduction	1
1.1 Project Motivation and Objectives	1
1.2 Project Approach	2
1.3 Structure of the Report.....	3
1.4 Scope of the Project.....	4
1.5 Important Terms and Definitions	4
Chapter 2 Background	5
2.1 Definition of Mobile Payment	5
2.2 Mobile Payment Adoption.....	5
2.3 Basic Four Corner EMV Payment Model.....	6
2.4 Tokenisation Applied in EMV Transactions.....	7
2.4.1 Anatomy of a Credit Card Number.....	7
2.4.2 Token Classification.....	8
2.4.3 EMV Tokenisation – Payment Token Ecosystem.....	8
2.4.4 EMVCo Tokenisation – New Data Elements.....	9
2.4.5 Primary Problem Solved by Tokenisation.....	10
2.5 Limited Use Keys and Cryptograms in EMV Transactions.....	10
2.6 Role of Meta Data in Fraud Prevention and 3-D Secure 2.0.0	13
Chapter 3 eWallet Solutions	14
3.1 Apple Pay–Embedded SE	15
3.2 Android Pay - HCE.....	16
3.3 Samsung Pay TEE - HCE.....	18
3.4 Why Apple Pay is Analysed in Further Detail	20
Chapter 4 Threat Discovery–Mobile Payment Model	21
4.1 Definition–Threat Targets for our Generic Mobile Payment Model.....	21
4.2 Evaluation –Threats to Mobile Payment Model	22
Chapter 5 Apply Threats, Evaluate Controls and Vulnerabilities	25
5.1 Evaluate – Manual Enrolment Process	25

Table of Contents

5.2	Evaluation–Contactless Payment at PoS (CP)	31
5.3	Interpretation and Conclusion	34
Chapter 6	Network Analysis - Card Enrolment Apple Pay	37
6.1	Scope of Network Analysis	37
6.2	Setup Description	37
6.3	Enrolment Process - with SSL/TLS Interception–‘Failed’	38
6.4	Enrolment Process without SSL/TLS Interception–‘Success’	39
6.4.1	Enrolment Process – User View	39
6.4.2	Enrolment Process–Network View.....	40
6.5	Analysis-Network View	41
6.6	Interpretation of Results.....	43
Chapter 7	Apple Pay and DSRP–How to Improve CNP.....	44
7.1	Known Weaknesses of CNP Transactions	44
7.2	Apple Pay–Digital Secure Remote Payment	44
7.3	How Apple Pay’s Approach Could Improve the Security of CNP Transactions	46
Chapter 8	Conclusion and Future Work	47
Chapter 9	Bibliography.....	51
Chapter 10	Appendix	56
	Definitions and Abbreviations.....	56
10.1	Components and Tools Used in this Work.....	58
10.2	Network Analysis - Screenshots.....	59
10.2.1	Http Trace – Web Proxy	59
10.2.2	DNS Resolution of Services	61
10.2.3	Location Overview.....	62
10.3	Various Screenshots	63
10.3.1	Access to Card Data via NFC Interface	63
10.3.2	Payment Receipts using Apple Pay at PoS Contactless	64
10.3.3	Transaction History using Apple Pay at PoS Contactless	65
10.3.4	Apple Pay at PoS Contactless ACR 123 Reader.....	66
10.3.5	Use of tokenPAN in CNP Transactions	67
10.3.6	Apple Pay–Payment Sheet	68

List of Figures

Figure 2:1 4 Corner Payment Model	6
Figure 2:2 Credit Card Anatomy.....	7
Figure 2:3 Token Taxonomy.....	8
Figure 2:4 EMV Token Ecosystem	9
Figure 2:5 EMV Payment Phases.....	10
Figure 2:6 EMV Session Key Derivation	11
Figure 2:7 EMV ARQC Generation.....	12
Figure 2:8 Meta Data, Contactless–3-D Secure 2.0.....	13
Figure 3:1 Apple Pay Ecosystem	15
Figure 3:2 Android Pay Ecosystem.....	17
Figure 3:3 Samsung Pay Ecosystem	19
Figure 4:1 Overview of considered stakeholders in threat model.....	21
Figure 5:1 Apple Pay–Overview of Card Enrolment Steps	25
Figure 5:2 ID&V Methods	26
Figure 5:3 Apple Pay – Manual Card Enrolment	27
Figure 5:4 Apple Pay–Contactless Payment PoS.....	31
Figure 5:5 Mobile Payment–Layered Security–Attack Targets and Controls.....	35
Figure 6:1 Network Analysis–Apple Pay Card Enrolment	37
Figure 6:2 Network Analysis, Apple Pay Card Enrolment, Failed with SSL Interception	38
Figure 6:3 Network Analysis of Apple Pay Card Enrolment-Success.....	40
Figure 7:1 Apple Pay–Web Payment DSRP	45
Figure 7:2 Apple Pay Web Payment–PaymentToken to Apple Pay.....	45
Figure 7:3 Apple Pay – Web Payment–PaymentToken to Merchant	46
Figure 8:1 Future Project – Introduction of IDE	48
Figure 8:2 Future Project–Introduction of Device Independency.....	49
Figure 10:1 Network Analysis–HTTP Trace	59
Figure 10:2 Network Analysis–DNS Output	61
Figure 10:3 Network Analysis–Global Service Distribution.....	62
Figure 10:4 NFC Android Card Reader	63
Figure 10:5 Apple Pay Receipts.....	64
Figure 10:6 Apple Pay–Transaction History	65
Figure 10:7 Apple Pay–ACS Reader Application Acceptance	66
Figure 10:8 Apple Pay–Use of Device Account Number in a CNP Transaction	67

Figure 10:9 Apple Pay—Payment sheet in a CNP Transaction68

List of Tables

Table 1:1 Important Terms and Definitions	4
Table 2:1 Important Data Elements of EMVCo Tokenization Specification	10
Table 2:2 Definitions for Cryptogram and Key Derivation	11
Table 2:3 Security properties ARQC.....	12
Table 3:1 eWallet Solutions-Overview	15
Table 4:1 Threat Targets	21
Table 4:2 Threats to the cardholder	22
Table 4:3 Threats to the smartphone	22
Table 4:4 Threats to the wallet and payment application	23
Table 4:5 Threats to the merchant	23
Table 4:6 Threats to the payment service provider	23
Table 4:7 Threats to the token service provider (TSP)	24
Table 4:8 Threats to the issuer	24
Table 4:9 Threats to the wallet service provider (TSM)	24
Table 5:1 Threats to the Cardholder–Details	27
Table 5:2 Threats to the Cardholder–Control Measures	28
Table 5:3 Threats to the Smartphone–Details	28
Table 5:4 Threats to the Smartphone–Control Measures	28
Table 5:5 Threats to the Wallet and Payment Application–Details	29
Table 5:6 Threats to the wallet application–Control Measures	29
Table 5:7 Threats to the TSP–Details	30
Table 5:8 Threats to the TSP–Control Measures.....	30
Table 5:9 Threats to Issuer–Details.....	30
Table 5:10 Threats to the Issuer–Control Measures	30
Table 5:11 Threats to the Wallet Service Provider–Details.....	31
Table 5:12 Threats to the Wallet Service Provider–Control Measures.....	31
Table 5:13 Threats on Smartphone–Details.....	32
Table 5:14 Threats to the Smartphone–Control Measures	32
Table 5:15 Threats to the Wallet Application–Details	32
Table 5:16 Threats to the Wallet Application–Control Measures.....	32
Table 5:17 Threats to the Merchant–Details	33
Table 5:18 Threats to the Merchant–Control Measures.....	33
Table 5:19 Threats to the TSP–Details	33

List of Tables

Table 5:20 Threats to the TSP–Control Measures.....33

Table 5:21 Threats to the Issuer–Details33

Table 5:22 Threats to the Issuer–Control Measures.....34

Table 6:1 Setup Network Analysis.....37

Table 6:2 Network Analysis–Connections.....41

Table 6:3 Network Analysis–DNS Resolution Overview.....42

Table 6:4 Network Analysis–Overview Results43

Table 7:1 CNP Transactions–CNP Weaknesses44

Table 10:1 Definitions and Abbreviations57

Table 10:2 Components and Tools.....58

Chapter 1 Introduction

1.1 Project Motivation and Objectives

In Switzerland, where I am presently living, the discussion surrounding mobile payment solutions has been ongoing for some time. During this time, I was employed as a contractor in a card issuing company (PCI DSS), where I was confronted with many legacy applications and a reluctance to adopt new mobile payment technologies without any clear reason. This was approximately one and a half years ago. This occurred in the context that point of sales terminals that accept contactless payments are being widely deployed. Contactless payments in Europe recently passed a major milestone, as Visa Europe [1] announced that three billion contactless transactions were carried out in the previous 12 months—nearly tripling the figure for the same period in the previous year of 2015.

Eventually, the situation in Switzerland has changed. A few months ago, Apple confirmed that it will cooperate with two minor card issuers in Switzerland, and on the 19th of October 2016 one of the larger card issuers announced its support for Apple Pay. Based on these recent announcements, I became curious, and decided to have a closer look at Apple Pay's ecosystem and its implemented security measures. The outcome may prove to be interesting, as Apple has a wide reputation for its commitment to privacy and security.

This project has the following objectives:

- A. The definition of a comprehensive model of the Apple Pay mobile payment ecosystem, which shows its integration into the current cache-less card payment model governed by the EMVCo/PCI DSS standards. This model will later be used for threat modelling.
- B. The application of threat modelling to evaluate what can go wrong, and how Apple has addressed these threats in their design.
- C. The methods Apple uses to address various threats shall be compared with the implemented security measures of other wallet solutions, to allow for a better understanding of Apple Pay's approach to security.
- D. The project shall provide a diagram, which visualises and provides insights into how the iPhone wallet communicates with the *Apple Cloud* during card enrolment and CNP payment actions.
- E. At the end of the report, I will outline how Apple Pay's approach to mobile payment could be beneficial for card not present (CNP) transactions in an online environment.
- F. In addition, the project work shall provide the reader with a reasoned justification regarding how effectively Apple Pay has applied the principle of tokenisation with respect to EMVCo and PCI Security Council guidelines, and the importance of tokenisation to Apple Pay and other wallet solutions will be elucidated.

1.2 Project Approach

Apple Pay contactless and other similar payment solutions are based on the EMV tokenisation standard, and must be compliant with contactless payment cards at PoS contactless readers. The author began this project with a literature review of the following material in order to gain a stronger understanding of the topic.

- EMVCo books with respect to contactless payment [83, 84, 85], tokenisation [18], and key management [21]
- PCI Security Standards for Tokenization [10]
- API description for the digital enablement services of VISA [12] and MasterCard [19]
- Official developer information regarding Apple Pay, Samsung Pay, and Android Pay
- Various technical literature regarding contactless payment, and technical papers of the GSMA, NCF & Smartcard Alliance, Mobey Forum, and other consortiums. These represent long term players in the smartcard and NFC industry.

The next step was to build up the relevant background information required to understand how tokenisation, cryptograms, and the collection of meta data influence the mobile payment process and support fraud detection.

Then, the functionality of the ecosystems for the three considered mobile payment solutions were established in the context of mobile payment. This already allows the major differences between the three approaches to be observed.

Having gained an understanding of the different systems, a focus was placed on Apple Pay, and generates a data flow diagram that will be used to evaluate possible threats to the card enrolment and the contactless payment processes at PoS. STRIDE [66] was used to categorise the threats to stakeholders in a generic mobile payment process.

Then, potential threats to the Apple Pay ecosystem were considered, and weaknesses (of which none were found) and strengths were investigated, in comparison with Samsung Pay and Android Pay.

However, before the author enrolled their MasterCard credit card into the Apple Pay wallet, a network analysis of the traffic that could be captured using a web proxy infrastructure was performed. This test provided a deeper understanding of the ecosystem and the traffic flows involved.

Finally, Apple Pay's digital secure remote payment (DSRP) system was analysed in terms of its general security benefits for CNP transactions. This analysis also led to the given recommended future projects, which would also make the Apple approach available to others.

Note: Owing to the closed nature of the payment infrastructure and its stakeholders, which use stringent non-disclosure agreements, this work is based on publicly available information.

1.3 Structure of the Report

This report is structured into following sections:



1.4 Scope of the Project

The work in this project focuses on Apple Pay as a mobile payment solution, and where meaningful compares it with Samsung Pay and Android Pay to set it into context with these two other EMV-based solutions and their slightly different approaches to the smartphone as a contactless payment device at PoS. In addition, Apple Pay's approach to remote secure payments will be discussed and analysed regarding its general applicability to the CNP environment in order to secure payments using EMV cryptographic methods.

- All three 'pay' solutions shall be introduced, to provide a common understanding.
- In the section on threats and vulnerabilities, we focus on Apple Pay's card enrolment process and the contactless payment process (CP) at PoS.
- The remote payment process (CNP) is not part of the threat and vulnerability analysis.
- An in-depth assessment of the different mobile operating systems is outside of the scope of this work. However, we will point out how different security features enhance the posture of the individual payment solutions.
- The same applies to security evaluations of other stakeholders, such as the TSP or issuer. We will list threats related to the mobile payment process, but will not go into a deeper analysis. A factor in this is that most of the relevant information is not available to the public.

The three chosen wallet providers match each other's functionalities in terms of payment and loyalty options, which the consumer is interested in. At the time of writing, the features regarding in-app and secure online payments change almost weekly. Because the contactless interfaces remain the same, CP transactions and card enrolment have been selected for analysis.

1.5 Important Terms and Definitions

For this report, we use following important terms and definitions. Additional definitions and abbreviations are listed in Chapter 0 .

Word, Expression	Description
acquirer	Acquirer bank, acquiring bank.
cardholder	The legitimate owner of the credit card including the digitised one in the eWallet.
contactless	This term is used for the contactless payment process at PoS using NFC technology.
credit card	This also includes debit cards, or just 'cards'.
CNP	Card not present transactions—typically remote transactions via the internet.
CP	Card present transactions - typically at PoS, where the cardholder presents the credit card.
issuer	Issuer bank, issuing bank.
payment device	The smart phone holding the digitised credit card in its wallet.
PAN	Personal account number printed on the front of the credit card.
tokenPAN	Tokenised (digitised) PAN used as surrogate PAN in payments.
TSP	The token service provider is responsible for tokenisation process.
sensitive payment data	We use the definition of the European Central Bank [62], "where sensitive payment data is defined as data that could be used to carry out fraud. These include data enabling a payment order to be initiated, data used for authentication, data used for ordering payment instruments, or authentication tools to be sent to customers, as well as data, parameters, and software which, if modified, may affect a legitimate party's ability to verify payment transactions or control the payment account."
smart phone	This includes mobile phones, digital assistants, iPhones, Samsung Galaxy, etc.
meta data	Meta data provides additional information about the subject, e.g., location information, device name, mobile number.
eWallet	The wallet or eWallet holds the digitised credit card used for payments, and is part of the mobile phone.

Table 1:1 Important Terms and Definitions

Chapter 2 Background

2.1 Definition of Mobile Payment

Dahlberg, Mallat, Ondrus and Zmijewska [5] defined mobile payments as *“payments for goods, services, and bills with a mobile device (such as a mobile phone, smart-phone, or personal digital assistant (PDA)) by taking advantage of wireless and other communication technologies. Mobile devices can be used in a variety of payment scenarios, such as payment for digital content (e.g., ring tones, logos, news, music, or games), tickets, parking fees and transport fares, or to access electronic payment services to pay bills and invoices. Payments for physical goods are also possible, both at vending and ticketing machines, and at manned point-of-sale (POS) terminals.”*

In this project, we focus on three mobile payment solutions based on using mobile phones as contactless payment instruments at PoS, and analyse Apple Pay’s security contribution with respect to remote payment transactions, i.e., CNP.

2.2 Mobile Payment Adoption

The widespread adoption of mobile phones has led to the emergence of innovative mobile services. Some of the emerging mobile services include a variety of banking and financial solutions, such as mobile payments, mobile microfinance, mobile vouchers and loyalty cards, and mobile banking. Studies based on the TAM (technology adoption methodology) have shown that the adoption of new payment technologies depends on compatibility, perceived usefulness, interconnection, perceived security, perceived ease of use, and payment habits. As these factors vary from one country to another, the success of a mobile payment solution in one country is not applicable in different countries [4]. Applying this theoretical background to the adoption rate of Apple Pay, Android Pay, or Samsung Pay payments shows that their possible success [4] is on account of a strong association with the EMV secure system, which is used as the baseline architecture in electronic and mobile payments for purchases. This approach fulfils the requirement of compatibility with the current PoS infrastructure and the perceived feeling of security, as it is based on available technology. From my point of view, the perceived usefulness, perceived ease of use, and compatibility with the payment habits of users will strongly influence the further adoption rate and success of these solutions.

The adoption rate will also be influenced by the perceived security of the payment method used, whether it is Apple Pay or another mobile payment solution.

Besides the TAM success factors, the solution will also have to comply with legal requirements. For example, the European Central Bank [6] has published a set of recommendations for making internet payments more secure, as surprises in the monetary system are undesirable for governments and central banks. However, because Android Pay, Apple Pay, and Samsung Pay all build on existing credit card infrastructure, and implicitly on the credit theory of money, they fit well into the current payment system.

2.3 Basic Four Corner EMV Payment Model

In order to better understand the business dynamics of the mobile payment ecosystems discussed in this paper, this section begins with an introduction into the *Four Corner EMV Payment Model*, used for traditional card payment systems. This has been highly influential in the development of the current global e-cash payment infrastructure. As a starting point, we will introduce the stakeholders, and illustrate the transaction flows in between the different participants. The model will be gradually extended during the project, to include the additional stakeholders present in the different payment ecosystems of Apple Pay, Samsung Pay, or Google's Android Pay.

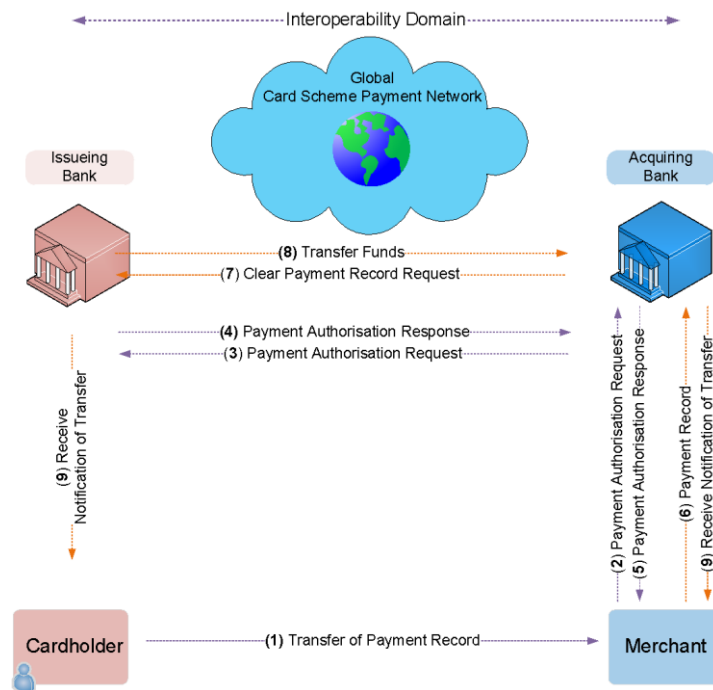


Figure 2:1 4 Corner Payment Model

Cardholder: The party that is considered as the end user in an NFC ecosystem. Essentially, the consumer is the user of the service who registers their credit card details with the service provider. They are responsible for initiating payment requests and agreements.

Merchant: Considered as the consumer matching part. The merchant offers products and services to consumers, and decides which payment options the consumer can use to make a payment.

Acquiring Bank: The main role of the acquirer is to handle financial payments by clearing and settling transactions through the financial institutions.

Issuing Bank: The cardholder is a customer of the issuing bank that issued the credit card.

Interoperability Domain: Consists of the global payment infrastructure supporting the payment process. This includes access control servers, payment processors, token service providers, and network connectivity, everything that is needed for the four main participants to be interconnected and process payments.

Payment flow explained:

1. The cardholder transfers the payment record, which includes the credit card number and agreed purchase amount, to the merchant.
2. The merchant creates a payment authorisation request, and forwards it to the acquiring bank.
3. The acquiring bank validates the merchant and the payment authorisation request, and forwards this via the payment network to the issuing bank.
4. The issuing bank validates the payment authorisation request, and further applies fraud analytics and other risk management procedures to minimise fraud. Following the completion of all verification processes, the issuing bank sends an authorisation response to the acquiring bank.
5. From there, the response is forwarded to the merchant.

6. In the case that the merchant receives a positive answer, the merchant will send the initial payment record from step 1 to the acquiring bank to initiate the payment clearing.
7. The acquiring bank generates and sends a clear payment request to the issuer bank to receive the funds.
8. The issuing bank transfers the funds to the acquiring bank.
9. The merchant and the cardholder receive a notification of the successful fund transfer.

2.4 Tokenisation Applied in EMV Transactions

One of the major challenges in the payment industry is to protect the primary account number (PAN) from being disclosed to prevent cross channel fraud. Tokenisation is not a new concept. It has been applied in the industry for quite some time as a mechanism for protecting payment credentials at rest against fraud and counterfeiting, by substituting the high-value payment credentials with a unique surrogate low-value equivalent. This protection of PAN data at rest within the card data environment (CDE) is regulated by the PCI Security Standards Council [9, 10]. In many cases, tokenisation allows for a reduction in the scope of PCI DSS, which can be a major cost reducing factor. For financial transactions in motion, the current tokenisation architecture is specified by EMVCo [18], where the token replaces the PAN with a substitute token value.

Owing to the global migration towards EMV payments at point of sales, where the payment transaction is now well protected from fraud by EMV chip technology, the European Central Bank [6] and others [61, 63] expect that CNP cross-channel and cross-border fraud will significantly increase. The use of tokenisation technology may become an efficient control, not only for protecting mobile payment solutions, but also as a general measure to tackle the impact of data compromises within CNP payment transactions.

2.4.1 Anatomy of a Credit Card Number

The credit card number assignment process follows ISO/IEC 7812-1:2015 ((*BS ISO/IEC 7812-1:2015: Identification cards. Identification of issuers. Numbering system*) [14]).

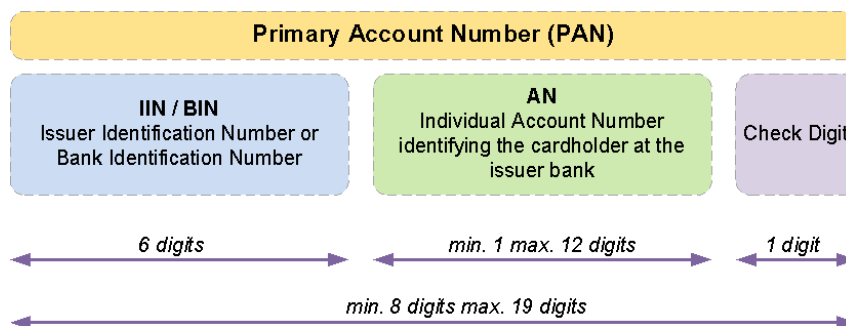


Figure 2:2 Credit Card Anatomy

IIN / BIN: The first six digits of the credit card number uniquely identify the issuer bank and the card scheme. The number serves as a label, and facilitates the transaction routing through the payment network to the issuer. This routing requirement is one reason why we require the BIN in clear text, rather than being encrypted.

AN: This is the individual account number, sometimes called the PAN, issued by the bank to uniquely identify the account holder. This is the most valuable part of the credit card number, as the other values are not confidential. A good tool for visualisation is the *Credit Card Validator* [15].

Check Digit: The last digit of the card number is the check digit, which is calculated by applying the Luhn algorithm [16] to the preceding parts of the credit card number. The check digit allows the fast validation of card numbers for typing mistakes, missing digits, and so on.

Note: During the tokenisation process, the original credit card number (PAN) will be replaced with a surrogate value or token. The token must have the same functionality as the original PAN to provide the required compatibility with the payment network. The token Service Provider (TSP) must assign a dedicated token BIN range to clearly separate original PANs from tokenised ones.

2.4.2 Token Classification

Tokenisation takes place at various parts of the payment process, and helps to protect sensitive PAN data during transaction processing, both at rest and while the data is in motion. Tokenisation in mobile payments constitutes the latter situation, and generates a *payment token*, which is used at PoS. Payment tokens are specified within the EMV and PCI specifications and guidelines, and are not mandatory to the stakeholders.

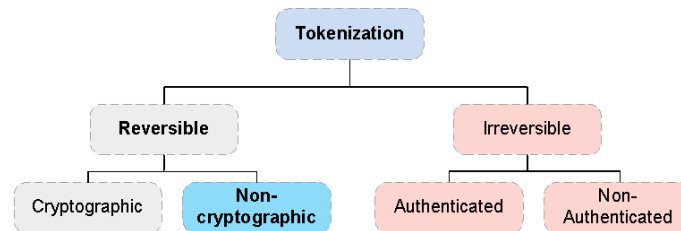


Figure 2:3 Token Taxonomy

(Derived from per PCI Security Standards Council [10])

TSP must generate *reversible tokens*, allowing the original PAN to be obtained from the token (de-tokenisation). We distinguish between tokens generated via *cryptographic* means and reversible *non-cryptographic* tokens, where we use a token lookup-table to retrieve the original PAN. *Non-cryptographic* tokens have no mathematical relationship with the original PAN value. TSP decides which method or combination to choose.

When these surrogate payment tokens are applied during an EMV transaction, a captured token cannot be related back to the original PAN and further used for cross channel fraud. The EMV Token Specification [18] details the specification, use cases, and the role model regarding tokenisation within the payment process.

As per the definition of PCI [10], tokens can generally be identified as either single-use or multi-use. A single-use token is typically used to represent a specific single transaction. A multi-use token represents a specific PAN, and may be used to track an individual PAN across multiple transactions. A multi-use token always maps a PAN value to the same token value within the tokenisation system. The determination of whether single-use or multi-use tokens, or a combination of both, are appropriate for a specific e-wallet solution depends on its specific needs. For example, in the case that an android phone is used as a payment instrument, where the payment tokens are hosted on a relatively insecure platform, we might consider single-use tokens, or introduce a method (HCE) to securely host single-use tokens but push one-time use payment cryptograms down to the android phone for in-time payments. This will be discussed further in Chapter 3.2.

2.4.3 EMV Tokenisation – Payment Token Ecosystem

The following diagram provides an overview of the various roles involved in the payment token ecosystem [18]. Some of these are existing roles in the traditional payments industry, and others are new, and introduced by the EMV Payment Token Specification [18]. The different roles will be analysed further as part of the threat modelling.

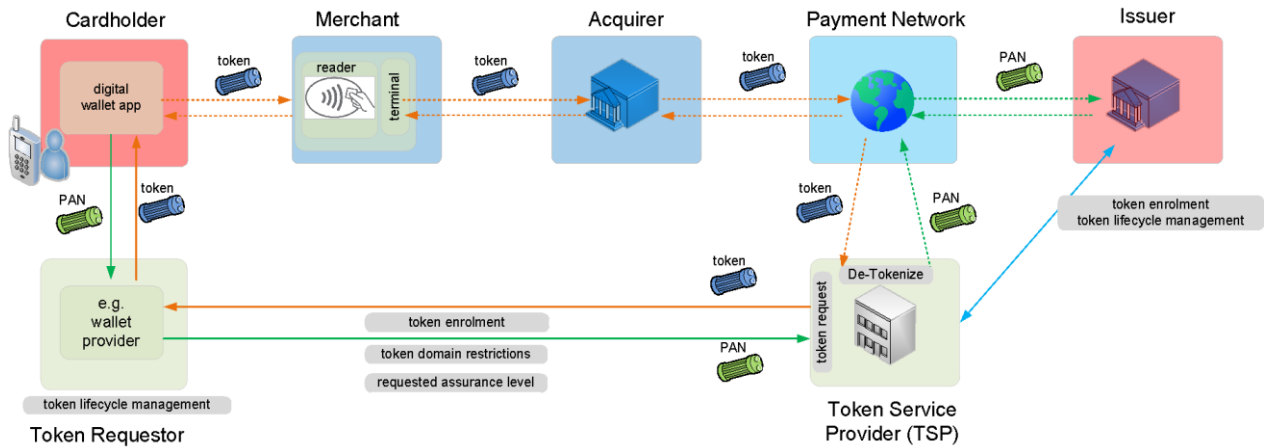


Figure 2:4 EMV Token Ecosystem

The new roles introduced in the tokenisation ecosystem are the *token requestor* and the *token service providers* (TSP) where the latter are authorised to provide payment tokens to registered token requestors. The TSP's main responsibilities are:

- operation and maintenance of a secure token vault
- key management of cryptograms
- provision of payment tokens, and management and allocation of non-overlapping BIN range
- enforcement of security and controls during tokenisation process
- identification and verification (ID&V) and token assurance
- assigning token domain restriction
- providing API for the token requesters and portal functionalities for issuers, e.g., MDES [19], VTS [12]
- real-time and continuous token lifecycle management (issue, re-issue, suspend, resume, delete, update, activate, expiry) can be triggered by different parties (token requestor, payment network, issuer), not only the issuer

2.4.4 EMVCo Tokenisation – New Data Elements

The main intention of the EMVCo tokenisation model is to limit the impact on fraud in the case of a data breach. The EMVC specification [18] adds the following new data elements, which may be employed in transactions using a payment token, and could help to limit and reduce possible fraud in the case that the payment token is disclosed to an adversary.

As shown in the table below, only a few of the data fields conform to a standard. All others are specific to payment networks, providing them with the required flexibility to implement their own interface. This is clearly reflected in the schemes' corresponding API interfaces. MasterCard calls this MDES [19], or the MasterCard Digital Enablement Service. The VISA equivalent of MDES is VTS [12], the VISA Token Service.

Data Element Name	Standard	Description
Payment Token	ISO 8583	Maintained by the token service provider. Tokens are assigned within an assigned BIN range to allow routing through the payment network.
Token Requestor ID		This value uniquely identifies the pairing of the token requestor with the token domain. The value contains TSP id Token Requester & Domain. The second component depends on the entity requesting the token and the defined domain in which the token should be used. The specification of the domain allows of the use of a token during transaction processing to be restricted to the assigned domain, at the time when the token was requested.
Token Expiry Date	ISO 8583	Maintained by the token service provider. The token expiry date must not be equivalent to the PAN expiry date.
POS Entry Mode		This specification uses the POS entry mode field to indicate the mode through which the payment token is presented for payment.
Last 4 Digits of PAN		This data will be printed on receipts and will also be used for charge back purposes or disputes.
Token Assurance Level		The token assurance level is a value that allows the token service provider to indicate the confidence level of the payment token to PAN/cardholder binding. This is determined by the type of ID&V performed and the entity that performs it. The token assurance level is set when issuing a payment token, and may be updated if an additional ID&V is performed. It represents a two-digit value, ranging from 00, indicating that no ID&V has been performed

		for the payment token, to a value of 99.
Token Assurance Data		This data may optionally be passed to the card issuer as part of the authorisation request, and may contain additional meta data from the payment method, e.g., location data or the ID&V methods used.
Token Cryptogram		This cryptogram is uniquely generated by the token requestor to validate the authorised use of the token. The token cryptogram will be passed in the authorisation request and validated by the token service provider and/or the card issuer.

Table 2:1 Important Data Elements of EMVCo Tokenization Specification

The token assurance level and token assurance data play an important role during the token enrolment. The ID&V methods used during the enrolment allow for a trusted binding of the payment token to the original PAN of the cardholder. This facilitates the assurance required to support secure and reliable transactions. Assurance can be performed and provided by various instances of the token ecosystem, namely none, the token requestor, the token service provider, the issuer, or a third party. The combination of all applied assurance methods forms the overall (layered) assurance level, being called *defence in depth*. An example of a potential pitfall was shown in Apple Pay's initial 'yellow path' [8] vulnerability during the ID&V process. Here, the verification and identification process for issuers was not strong enough, and allowed fraudsters to register stolen credit cards. The card enrolment process will be further analysed in Chapter 5.1.

2.4.5 Primary Problem Solved by Tokenisation

Through the replacement, of the PAN with a surrogate value within a payment transaction, i.e., tokenisation, we can remove sensitive account data from the payment environment. In the case of a data breach, a tokenised PAN is of very little value outside of the assigned token domain. Therefore, tokenisation can work together with other methods, such as end-to-end encryption or EMV chip technology, to reduce fraud. According to Verizon DBIR 2016 [11], point of sales malware (ram scrappers) retrieving account information are still a reliable origin for stolen payment data. Tokenisation would help to reduce the impact of such data breaches and limit their possible use.

As previously mentioned, one key characteristic of payment tokens is the fact that a token is connected to a domain restriction, which is assigned during the token request and token provision process. This domain restriction is part of the token requester ID. The domain restriction allows the use of the payment token to be bound to an assigned device, a specific channel (e.g., contactless), or a merchant shop. These all depend on the restrictions assigned during the token request procedure. The successful application of domain restrictions allows the issuer to take necessary measures to prevent cross channel fraud stemming from re-use of the stolen payment details.

Tokenisation introduces new vulnerabilities, as well as reducing the likelihood of well-known ones. The new stakeholders also broaden the attack surface. The 'yellow path' [8] vulnerability is one example. Another high rank target is the TSP, which is where the token translation takes place.

2.5 Limited Use Keys and Cryptograms in EMV Transactions

Contactless mobile payment solutions, such as Apple Pay, must satisfy minimum compliance with the EMV specification 3.0 [21] and the corresponding contactless reader specification of the scheme [20]. Therefore, mobile contactless payments use standard a transaction authorisation method (e.g., ARQC, ARPC). Below, the standard phases of an EMV based payment process are illustrated.

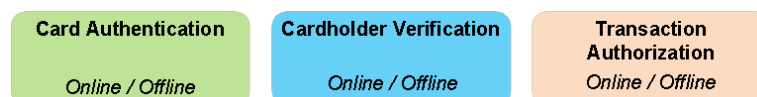


Figure 2:5 EMV Payment Phases

During an online transaction authorisation process, the EMV chip card generates an authorisation request cryptogram (ARQC) [21], which is sent via the payment network to the issuer for authorisation. Cryptographic mechanisms are applied to provide the required transaction integrity. The online authorisation or authentication tasks executed in an EMV chip use cryptographic keys,

which are shared with the issuer. In the case of offline tasks, the EMV chip contains asymmetric key material, which is used to sign data, and can be verified locally at PoS.

Besides maintaining the secrecy of the PAN by tokenisation, the main concerns of the payment industry with respect to the transaction process are the transaction integrity and data origin authentication, which provide the necessary evidence that the cardholder’s transaction has not been changed and that the cardholder’s card has been used. This is where the generation of the payment cryptogram is relevant.

In EMV chip-based credit card payment solutions, we can safely store cryptographic key material within the SE of the smart card. This facilitates the application of strong card authentication and payment authorisation mechanisms. This capability, and the need for secure storage for the cryptographic key material, constitute major requirements for mobile payment solutions. However, mobile phone platform providers do not always provide secure elements (SE) or trusted execution environments (TEE) to guarantee the same security as an EMV chip card. As a mitigation method, EMV [21] incorporates the concept of deriving dynamic cryptographic key material, which facilitates the use of different keys for each payment transaction. In the case that an attacker can eavesdrop on payment transactions, this prevents the captured data from being used for further reply attacks, i.e., cross-contaminations.

The derivation of the dynamic key material (session key) used in an authorisation cryptogram follows a certain pattern in an EMV transaction, which is described below.

Definition	Description
UN	Random number generated by the PoS terminal
MAC	Message authentication code algorithm used to provide integrity
DDOL	Dynamic data authentication data object list
amount	Amount payable
Currency	Currency code
Exp Date	Expiry date
ATC	Application transaction counter, incremented after each payment
other data	Not specified as standard
DATA	Certain relevant customer data used as input for key derivation
tUDK	Unique derived token key for this customer
tokenPAN	Unique tokenised PAN for this customer
tPAN	Unique tokenised PAN for this customer
LUK	Derived limited use key for this transaction–session key

Table 2:2 Definitions for Cryptogram and Key Derivation

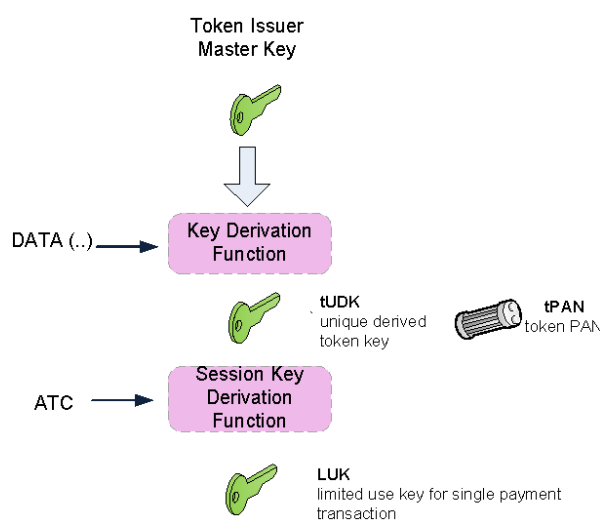


Figure 2:6 EMV Session Key Derivation

(Derived from EMVCo, Book 2 Key Management [21])

At the same time that the tokenised PAN (tPAN) is generated, a unique derived token key (tUDK) is also generated. Together with the tokenised PAN(tokenPAN), this key is later transferred and stored either in a secure location on the mobile platform (SE, UICC, TEE) or in a similar manner in the cloud in case of HCE (host card emulation). The session key or limited use key (LUK) is generated at the run time, when there is a requirement for a session key during the authorisation cryptogram generation.

The generated ARQC consists of clear text terminal data, containing information regarding the purchase and bank-specific data, including the tokenised PAN and a message authentication code (MAC), which is generated for the terminal data and bank specific data.

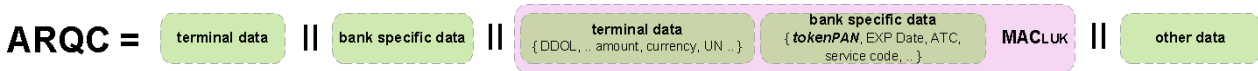


Figure 2:7 EMV ARQC Generation

(Derived from EMVCo, Book 2 Key Management [21])

The ARQC contains clear text fields that are integrity protected. The message authentication code allows the validation of the cryptogram, either by the TSP or the issuer. Note that the EMV does not encrypt data, but rather it authenticates the data.

The security properties of ARQC are described in the following:

Security Property	Description
Data Integrity	Data integrity of the payment data is protected by generating the MAC
Data origin authentication	Data origin authentication can be verified via MAC
Reply protection	Reply protected – (UN, ATC) MACLUK as ATC changes as well as LUK and makes reply attacks much harder. This works in tandem with the tokenPAN's domain restriction.
No confidentiality	no confidentiality between PoS reader and payment device – all payment data is available in the clear
No_ Non-repudiation	Because we use shared key cryptography to generate the MAC, we cannot guarantee non-repudiation regarding the cardholder as more than one party is in possession of the shared key (LUK)

Table 2:3 Security properties ARQC

The payment cryptogram generated during the mobile payment process is very similar to the ARQC, but additionally contains unique authentication and meta data generated by the smartphone device. The payment cryptogram demonstrates to the card network that the device and card being used are genuine, and not a vehicle of intercepted or cloned credentials.

Note: The payment cryptogram depends on the tokenPAN, ATC, and other transaction-specific data. Even in the case that the tokenPAN is disclosed, this is worthless for a fraudster, as the required cryptogram (MAC) cannot be generated.

2.6 Role of Meta Data in Fraud Prevention and 3-D Secure 2.0

In the payment process, the stakeholders mainly focus on providing service availability, interoperability, and non-repudiation. The latter is connected to fraud prevention. To secure the payment process, we successfully apply methods such as tokenisation, secure element technology, cryptography, and other techniques. Going on step further, we require methods to *detect* fraudulent activities, and eventually we must *respond* to and *recover* from fraudulent activities. This methodology is further detailed in NIST's Cyber Security Framework [67], and often deployed in enterprise infrastructures to address security aspects.

The introduction of smartphones as payment devices introduces the potential to collect and access a large range of meta data. At the time when Apple Pay was introduced, privacy issues were a major concern, even though Apple's privacy policy [31] stated that no sensitive payment information was stored on their side.

The seamless integration of Apple Pay at PoS for contactless payment does not allow additional meta data to be incorporated to support further fraud analytics. The structure of the payment packet must fit into the current PoS infrastructure using contactless cards. Nevertheless, there is additional data available, which is collected during *enrolment* and *transaction* processing, as illustrated in the image below. This data can be helpful for differentiating between legitimate cardholders and fraudsters.

The available meta data for advanced fraud analytics are listed below. The number of values varies from platform to platform. The difference might depend on the security measures of the platform, which do not allow the retrieval of certain confidential values.

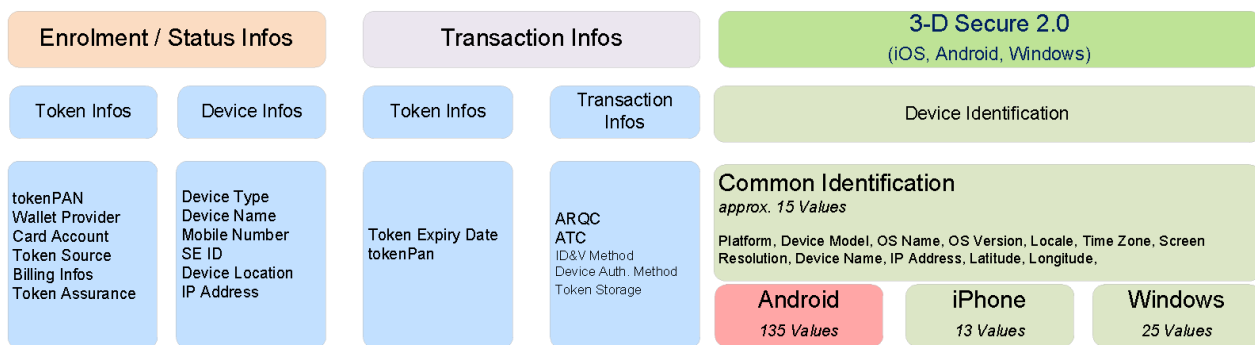


Figure 2:8 Meta Data, Contactless-3-D Secure 2.0

Owing to the privacy concerns regarding the collection of meta data, the stakeholders, including the cardholder, must agree on how much information is required to prevent fraud in accordance with their risk appetite, while remaining compliant with data privacy laws in the relevant jurisdiction, e.g., the Data Protection Act 1998 in the UK. The amount of additional information collected in the 3-D Secure 2.0 device information list [81] indicates the direction that is taken. It is attempted to fingerprint the device and the cardholder, and to establish a usage profile. The introduction of *3-D Secure's Software Development Kit (SDK)* for mobile applications confirms that EMV focuses on the mobile payment and the frictionless integration of this with 3-D Secure authentication and verification. The SDK fits very well into the current mobile payment infrastructure, and constituted an important missing component before the end of 2016.

In my opinion, Apple and other eWallet providers do well in transferring this 'data collecting task' to the stakeholders of the payment industry to avoid crossing any legal boundaries in terms of privacy regulations.

Chapter 3 eWallet Solutions

In this chapter, we will introduce the various ‘pay’ wallet solutions. Apple Pay, Android Pay, and Samsung Pay are currently the only mobile wallet models [61] in the marketplace that follow the EMV specification, requiring payment tokenisation and issuer ID&V [18]. They are compliant with the EMV contactless card standard ISO/IEC 14443(a|b), and can be used as a payment method, where supported.

In addition, Samsung Pay also supports MST (magnetic secure transmission) as well as NFC contactless compliance. MST allows the use of contactless payment at standard magnetic stripe terminals. This would have been a significant factor in enabling this method’s adoption before the US began migration towards EMV chip technology, and began updating their infrastructure to achieve EMV compatibility, including PoS readers.

The table below provides an overview of the three ‘Pay’ solutions and their relevant characteristics.

Solution	Vendor	Operating System	Description of relevant characteristics
Apple Pay <i>embedded SE</i> 	Apple	iOS	<ul style="list-style-type: none"> - high value tokenPAN and cryptographic keys are stored/strongly protected in a secure element (SE) - Issuer certified payment app is stored in SE - SE is managed by Apple - use of scheme controlled (TSP) tokenisation—multi-use token - use of scheme controlled (TSP) cryptographic key material - static - NFC interface is only accessible via SE and not open for other apps - only Apple Pay wallet application can be used for payment - Apple Pay does not need internet connectivity during payment process - fingerprint and PIN based CV
Samsung Pay <i>Trusted Execution Environment (TEE)</i> 	Samsung	Android Galaxy Platform	<ul style="list-style-type: none"> - high value tokenPAN and cryptographic keys are stored and well protected in the trusted execution environment (TEE), called TrustZone, which is part of the Samsung KNOX Security Framework - issuer certified and trusted payment application reside in TEE - use of scheme controlled (TSP) tokenisation—multi-use token - use of scheme controlled (TSP) cryptographic key material—static or dynamic depending on card issuer or brand scheme - Samsung Pay’s user interface resides in rich OS - NFC interface and communication to payment app is controlled by TEE’s trusted drivers - besides NFC, Samsung Pay does also support MST contactless - Theoretically, Samsung phones can also run Android Pay, which runs the same OS. In the case that Samsung KNOX is installed; Android Pay cannot be installed [45].
Android Pay <i>Host Card Emulation SE in the Cloud</i> 	Various	Android	<ul style="list-style-type: none"> - high value PAN and cryptographic keys are stored and protected in Google’s cloud environment, SE in the Cloud - payment application resides in rich OS, neither in a TEE nor in SE technology-based storage - dynamic/limited use payment credentials including tokenPAN reside within a dedicated storage area, protected via cryptographic means from unwanted disclosure (white box cryptography) - use of scheme controlled (TSP) tokenisation—multi-use token - use of scheme controlled (TSP) cryptographic dynamic key material - host card emulation is open to any application—NFC reader access is not specifically protected - issuer can freely develop their payment applications—open environment
Generic SIM centric <i>simSE/ UICC</i>	Various	Various	<ul style="list-style-type: none"> - high value tokenPAN and cryptographic keys are stored/strongly protected in UICC - secure element (SE) - issuer certified payment app is stored in UICC chip - SE is managed by MNO or TSM - use of scheme controlled (TSP) tokenisation—multi-use token - use of scheme controlled (TSP) cryptographic key material - static - NFC interface protection varies


			<ul style="list-style-type: none"> - issuers are free to provide their payment applications - owing to the many stakeholders, the ecosystem has an increased complexity <p>Owing to this similarity with Apple Pay, which uses embedded SE, we will not consider this solution in further detail, despite the fact that in some countries where the maturity of MNO and their cooperation with banks works seamlessly and efficiently this may be the fastest approach to getting the project up and running.</p>
	Info	contactless	Support for contactless mobile payment is signalled with this symbol. Regarding support for individual mobile payment solutions, the corresponding icon will be shown as well.

Table 3:1 eWallet Solutions-Overview

To describe the solutions in the following chapters, we used ENISA’s categorisation given in its recent threat report [60] regarding mobile payment solutions. We describe card *enrolment*, *payment process*, *user authentication*, *device authentication*, and *data protection*. This allows the reader to understand how the different stakeholders and components work together, and where the main differences lie between Apple Pay, Samsung Pay, and Android Pay.

3.1 Apple Pay–Embedded SE

In this chapter, we introduce Apple Pay’s ecosystem. We dissect the system into the relevant components involved in the payment process, and list the external stakeholders. We will later use this model to evaluate possible threats to the ecosystem, and to investigate how well it is protected. The details of how the Apple Pay application interacts with system and external stakeholders are documented in the iOS security guide [34].

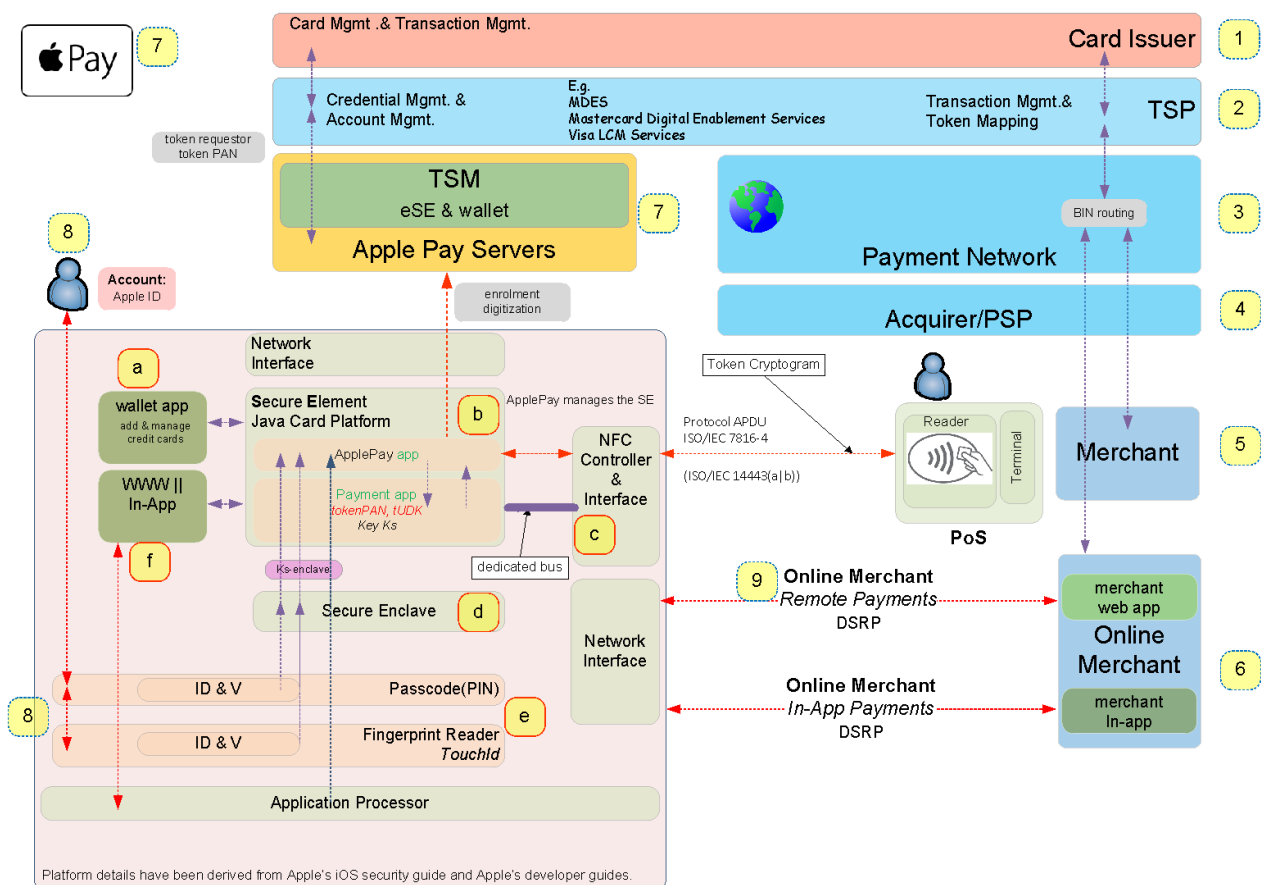


Figure 3:1 Apple Pay Ecosystem

Platform: Apple Pay’s contactless payment method is supported on the platforms of the iPhone 6 and higher. It can be used for contactless EMV payments at PoS (c,5), as well as for digital secure remote payments (DSRP). These payments are commonly described as in-app payments [32]. CNP payment transactions benefit from EMV-like transactions, in contrast to common CNP remote payments, where 3D-Secure is used to strengthen the cardholder verification and identification process. In-app payments can also

be conducted via Apple's iPad or OSX devices in the case that they comply to the minimum hardware and software requirements [35].

User Authentication: Apple Pay's ID&V process **(e)** for the cardholder enforces biometric fingerprint verification, called *Touch Id*. After five unsuccessful attempts [34] to match a fingerprint, the device allows for passcode authentication. Note: Practical tests in the lab have shown different numbers. Only after successful authentication, whether authenticated with fingerprint or using passcode as fall-back method, can a payment transaction be authorised. The scanned fingerprint is temporarily stored in the secure enclave **(d)**. The secure enclave **(d)** communicates with the secure element (SE) via a shared pairing key (*Ks*), which is provisioned during the manufacturing process.

Data Protection: Apple Pay's SE **(b)** is an industry-standard, certified chip running the Java Card Platform [36], which is compliant with financial industry requirements for EMV payments [34], and there applied for contact and contactless payment transactions. Within the secure element structure, which is managed by Apple Pay **(7)**, the Apple Pay applets, certified issuer payment applets, tokenPAN (device account ID), and unique derived key (tUDK) are securely stored. The wallet application **(a)** is the user interface (UI), and manages and stores the payment cards, velocity cards, and so on. Apple does only allow the apple wallet to communicate with the secure element API. Apple Pay's in-app payment [32] system offers a dedicated API, which can be called by the web application to initiate a payment. Apple does not allow other applications to access the NFC **(c)** controller to emulate a credit card (HCE). The NFC does control the communication to the secure element, and owing to the restricted design, no payment data is available outside of the SE and NFC controller.

Card Enrolment: Apple Pay's manual card enrolment will be analysed in further detail later in this report. For further details, see Chapter 6. To enrol a credit card into the wallet **(a)**, the card issuer **(2)** must support and have an agreement [27] with Apple Pay **(7)**. The cardholder **(8)** has three options for adding a credit or debit card into the wallet application. First, the credit card can be added manually, where the card number is scanned or manually typed in, and the CVV number is provided as a method of cardholder verification. Importantly, Apple Pay does not store either the PAN or the CVV. These values are only used during enrolment as part of the ID&V procedure [31]. Second, in the case that the cardholder already has an iTunes account with a supported credit card on file, this credit card can be added into the wallet. Finally, the credit card can also be added using a card issuer's application. During enrolment, the Apple Pay server acts as the token requestor **(7)**, and send a request for the tokenisation to the assigned token service provider **(2)**. The token service provider (TSP) will contact the issuer to verify the enrolment eligibility of the cardholder, and to check whether additional ID&V steps are required. Finally, the credit card number (PAN) will be digitised (tokenised), and a unique shared key (tUDK) will be generated and eventually added to the SE for storage.

The **payment process** begins at the PoS contactless reader **(5)**. The cardholder presents their NFC interface **(c)** to the *contact reader sign*, and identifies themselves as the legitimate cardholder and device owner by authenticating with their fingerprint (Touch Id), which also authorises the payment. Importantly, the wallet always asks for fingerprint authentication. Apple's fingerprint authentication method is called consumer device cardholder verification (CDCVM) in EMV terms, and allows Apple Pay to function without floor-limits in most countries [28]. After the transaction has been authorised, the payment applet **(b)** generates the authorisation request cryptogram (dynamic transaction data), which the merchant **(5)** forwards via the acquirer **(4)** and payment network **(3)** to the TSP **(2)**. The TSP validates the authorisation request cryptogram, de-tokenises the tokenPAN into the original PAN, and forwards the PAN and payment data to the issuer **(1)** for fraud management and the final authorisation of the transaction.

Note: Apple Pay's extension of in-app payments to iPad and Mac OSX devices and its relevance for future CNP payments will be discussed in Chapter 7.

3.2 Android Pay - HCE

Because Android Pay is completely software-based and is still in the early adoption phase, the technology and architecture used may still be changed. At some stage, Google decided to declare the Android phone as compromised for use as a payment method, and introduced host card emulation (HCE) [7, 40], together with EMV's tokenisation standard and dynamic cryptograms. Instead of generating and storing high-value payment credentials on the phone, this will be stored in the Google Cloud.

In comparison with Apple Pay and Samsung Pay, where card emulation is restricted to highly secure areas using SE and TEE technology, Google decided to abandon their SE technology, and by the end of 2013 re-introduced Blackberry's [42] secure card emulation, without a using an SE, on its Android 4.4 "KitKat" operating system. Google called this host-based card emulation (HCE), where here HCE allows any app to communicate APDUs **(c)** with the PoS Terminal. This option is useful in allowing service providers to

quickly create their own payment solutions. However, they need to be aware of the security risks resulting from the lack of hardware-based security isolation as provided by an SE or TEE. At the time of writing, the virtual SE is located in the Google Cloud.

Although cloud based technologies can function well with Android Pay, this is technically not the only way to enable Android Pay. TEE technology may provide another option to be leveraged by Android Pay, analogously to Samsung Pay. There have already been attempts to secure the fingerprint reader via TEE technology [37]. The final architecture will depend on the acceptance of the solution by payment schemes and issuers. However, as Android Pay uses the digital enabling services [19, 12] of the card schemes to implement their wallet solution, they are more likely to be accepted. This satisfies compliance with EMV standard facilitates open-loop card payments, like the two other solutions.

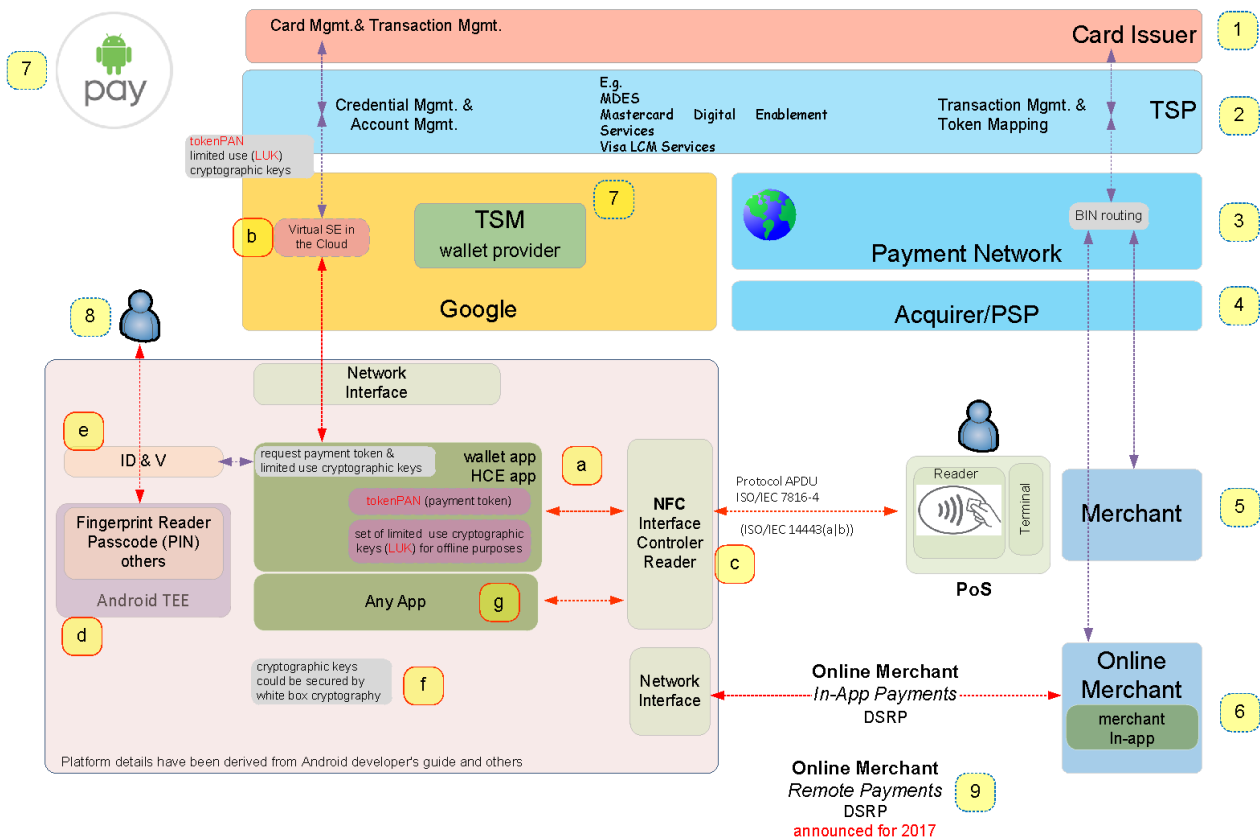


Figure 3:2 Android Pay Ecosystem

Platform: Android Pay’s contactless payment method requires at least Android 4.4 “KitKat” and an NFC reader interface (c). In comparison with Apple Pay [34], where only payment requests arriving from an in-field terminal are marked by the NFC controller as contactless transactions, Android’s NFC interface can be literally accessed by any program (g). Furthermore, it can be used in an active reader application [43] to read in an unauthorized manner contactless card details from other cardholder’s wallets. The impact of this data disclosure can be minimised through the use of tokenisation and dynamic cryptograms. Aside from this weakness, issuers benefit from this direct access to the NFC controller (c), and they can provide their own payment applications via the Google play store. While Apple Pay and Samsung Pay use established hardware-based security measures, such as the tamper-proof secure element (SE) or TEE, which have been researched and tested against the latest EMV focusing attack vectors, Android’s software-based solution and security has not been tested in a comparable manner. Hence, the current state of security is difficult to establish. There are tendencies to introduce TEE [37] to protect the fingerprint reader and other authentication (ID&V) data.

User Authentication: Android Pay’s ID&V process (e) for the cardholder supports biometric fingerprints, IRIS scans, standard PIN codes, or pattern based authentication to authorise payment transactions, where no authentication method is enforced. The cardholder can authorise a payment as soon as the phone is unlocked, according to the developer guide, and there is no additional transaction authorisation required [44]. Besides the possible application of TEE to protect authentication data [37], there is no further information available concerning the security protection mechanism applied to the authentication process.

Data Protection: Android Pay does not provide a secure and tamper-proof location for storing payment credentials. To mitigate the possible risk of a data compromise, Android Pay uses the so called virtual secure element (SE) in the cloud **(b)**, which is managed by Google, who assume the roles **(7)** of the TSM (trusted service manager) and the TR (token requestor) **(7)**. The wallet application must contact Google to receive the tokenPAN and the limited use keys (LUK). To prevent a negative user experience, the wallet application is pre-published with several limited use keys (session keys), which are pre-stored in the mobile OS to enable the transaction to be completed without network connectivity. Considering the key derivation process described in Chapter 2.5, these keys could be pre-generated for expected future ATCs, and will be processed in the same manner as an Apple Pay or Samsung Pay cryptogram. The storage of sensitive credentials could also be protected while using white-box cryptography, which implements cryptographic algorithms in software to render it difficult for an attacker to retrieve key material [46].

Because HCE solutions do not require the secure storage of payment credentials on the device, payment security is (must be) provided through the layering of multiple security solutions, to provide the same security provided by a hardware solution.

Card Enrolment: First, to enrol a credit card into the wallet **(a)**, the card issuer **(2)** must support and have an agreement [38] with Google's Android Pay **(7)**. Then, a credit or debit card can be registered with Android Pay. The only verification conducted during the enrolment process is the cardholder verification performed by the issuer, asking for the CVV or additional issuer defined ID&V attributes. These methods depend on the issuer's risk appetite, and include email, SMS, a phone call, 3-D Secure, or its own banking application. Therefore, Google delegates the ID&V process to the card issuer **(1)**. After registering the credit or debit card with Android Pay, Google acts as the token requestor for the TSP **(2)**. The TSP will contact the issuer to verify the enrolment eligibility of the cardholder.

Security and privacy: In comparison with Apple Pay, where no credit card details are stored, an Android Pay user transmits their credentials into the Google Cloud. After successful enrolment, the Android Pay wallet **(a)** is published with the tokenPAN **(b)** and several limited use payment credentials. In the case of a data breach, stolen credentials are of limited use. For details, refer to Apple Pay and tokenisation in Chapter 2.4.

The **payment process** starts at the PoS contactless reader **(5)**. The cardholder presents the NFC interface **(2)** to the *contact reader sign*, and identifies themselves as the legitimate cardholder/device owner by unlocking the device with the chosen authentication method – this also authorises the payment. Importantly, the wallet application does not request further authentication. After the cardholder has authorised the payment, the wallet app **(a)** employs the limited use keys (LUK) to generate the authorisation request cryptogram (dynamic transaction data), which the merchant **(5)** forwards via the acquirer **(4)** and payment network **(3)** to the TSP **(2)**. The TSP validates the authorisation request cryptogram, de-tokenising the tokenPAN into the original PAN and forwarding the PAN and payment data to the issuer **(1)** for fraud management and the final authorisation of the payment transaction. The wallet app is frequently replenished with new dynamic key material **(b)**, and therefore requires internet connectivity.

3.3 Samsung Pay TEE - HCE

Samsung Pay makes use of TEE to secure data, and the application platform is based on the KNOX framework [49]. The Samsung KNOX hardware platform creates two parallel execution environments with a strong segregation. One environment runs the standard non-privileged user application, and the other environment, called the trusted execution environment, runs privileged applications such as authentication mechanisms, applications for data encryption, and the payment application. Thus, KNOX [49] facilitates the necessary separation to protect sensitive data from attackers. All connections into the TEE are controlled by trusted APIs **(d)**. In the case that the TEE application needs to access externally located devices, this access is provided via trusted device drivers **(f,d)**.

Samsung Pay supports EMV tokenisation and can use static **(b1)** or limited use keys **(b2)**, integrating HCE in the latter case. This adds additional flexibility to fulfil different issuer requirements. One outstanding feature is Samsung Pay's support for the magnetic secure transmission protocol (MST). Magnetic secure transmission **(g)** emulates a magnetic stripe card (MSC), and can therefore be used at traditional and widely deployed MSC PoS terminals. This removes the dependency on contactless ISO/IEC 14443 support at PoS.

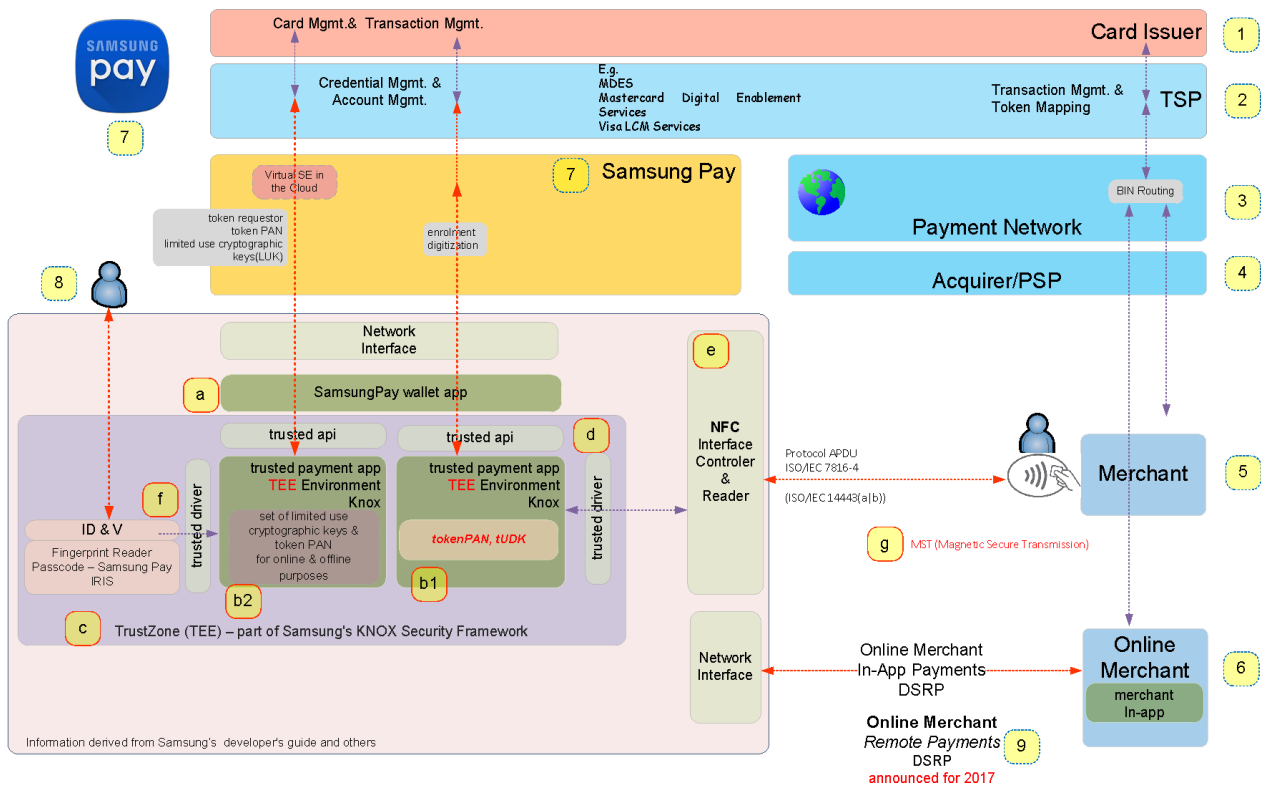


Figure 3:3 Samsung Pay Ecosystem

Platform: Samsung Pay’s contactless payment solution, applying KNOX [24] technology, is available on the latest smart phones (e.g., Galaxy S7) [54]. Analogously to Apple Pay’s SE and secure enclave, the KNOX platform helps to provide an isolated, secure platform within the main platform. The idea of using trusted computing technology to emulate EMV card technology has been present for some time [41]. The TEE environment is aimed towards achieving this. As with iPhone or UICC based solutions, the secure application delivery plays a crucial role in guaranteeing the necessary integrity for payment applications. Access into and out of the TEE for the payment application needs to be routed through the trusted APIs and drivers (d). For example, communication with the NFC controller (e) is restricted to the trusted NFC driver. In comparison with the embedded SE of Apple Pay, Samsung’s KNOX platform allows other trusted applications to benefit from this secure environment.

User Authentication: Samsung Pay’s ID&V process (f) authenticates the cardholder/device owner either by the fingerprint [48] scanner or the trusted PIN, both of which reside in the TrustZone (TEE). Every payment transaction must be authorised by the cardholder via authentication. Authentication results are encrypted with the trusted payment applications key, and are immediately cleared after transmission to prevent any single user authentication from being used to attempt multiple payments [49]. The cardholder can choose which authentication methods they would like to use, either PIN or fingerprint.

Data Protection: Samsung’s security-certified [51] KNOX (c) architecture provides a secure and tamper-proof location for storing payment credentials. Like Apple Pay, the KNOX environment makes extensive use of cryptographic mechanisms to secure communication between applications and device drivers within TEE. To further mitigate the possible risk of data compromise, Samsung Pay applies tokenisation [25, 50] with dynamic keys, which are replenished frequently, or static key material. This is similar to Apple Pay’s approach, but with TEE being used instead of SE technology. As a wallet provider and Token Requestor (7), Samsung Pay signs up with a token service provider (2) for tokenisation and credentials management. In the case that Samsung Pay stores the unique derived key within the TEE, it can generate the dynamic payment cryptograms within the TEE (see the key derivation process in 2.5), and does not require the use of HCE technology. In the case that the payment network requires dynamic keys, the cardholder’s device must obtain the keys before use, and these must be invisible to the cardholder. Multiple keys will be downloaded to the smartphone, with a replenishment process similar to Android Pay. In the case that the limited use keys are depleted or expired, the smartphone must access the internet and obtain new keys for future payment transactions. Again, this functions in the same manner as Android Pay’s HCE-based solution.

Card Enrolment: First, to enrol a credit or debit card into the wallet **(a)** application, the card issuers **(1)** must support and have an agreement [55] with Samsung Pay **(7)**. Samsung Pay requires a contractual agreement with the issuer's token service provider **(2)**. After the contractual requirements are met, the cardholder is eligible to add a credit card into the wallet. The cardholder initially provides the card number, name, expiry date, and CVV to identify themselves as the legitimate cardholder. Next, Samsung Pay's enrolment server **(7)** acts as the token requestor, and sends the request for the tokenisation to the assigned TSP. The TSP contacts the issuer for the cardholder's eligibility to enrol their card. Currently supported ID&V methods are [53] one time tokens, sent via SMS, email, or a bank call; app-to-app channels; or any other issuer-initiated 'out of band' authentication. After successful enrolment, the Samsung Pay wallet **(a)** is either published with the tokenPAN **(b2)** and several limited use payment credentials, or **(b1)** a unique derived key identical to Apple Pay. It is then ready to be used.

The **payment process** starts at the PoS contactless reader **(5)**. The cardholder holds his NFC interface **(2)** to the *contact reader sign* and identifies himself as the legitimate cardholder while unlocking the device with the chosen authentication method. Then the cardholder authorizes the payment via fingerprint or pin. There is no fingerprint authorization enforced. After transaction is authorized, the wallet app **(b2)** uses the limited use keys or in case of option **(b1)**, derives the limited use key based on the ATC (2.5) to generate the authorisation request cryptogram (dynamic transaction data). The payment cryptogram and the tokenPAN are sent via merchant **(5)**, acquirer **(4)** and payment network **(3)** to the TSP **(2)**. The TSP validates the authorisation request cryptogram, de-tokenize the tokenPAN into the original PAN and forwards PAN and payment data to the Issuer **(1)** for fraud management and final authorisation of the payment transaction. The Samsung wallet app running HCE option **(b2)** gets frequently replenished with new dynamic key material **(b2)** and therefore needs internet connectivity.

3.4 Why Apple Pay is Analysed in Further Detail

End of October 2016, when the project started, Apple Pay was the only mobile payment solution available in Switzerland, supported by card issuers and allowing open-loop payments and EMV like in-app [32] and web payments (DSRP) [23] in remote payment transactions. The initial enablers were MasterCard's digital enablement platform [19] and VISA's VTS [12] service.

Support for DSRP will be a major enabler in secure CNP transactions in the future. This will be further analysed in Chapter 7.3 and Chapter 8. The other two vendors, Android Pay [22] and Samsung Pay, support in-app payments, and recently announced corporation with Visa Checkout and Mastercard's Masterpass [52]. However, in my opinion Apple Pay's approach to DSRP could be used to design a non-proprietary solution. For further details, see Chapter 8.

Chapter 4 Threat Discovery–Mobile Payment Model

4.1 Definition–Threat Targets for our Generic Mobile Payment Model

The following picture shows the main stakeholders who will be analysed for threats and vulnerabilities. The dashed lines illustrate the trust boundaries where we analyse what could go wrong.

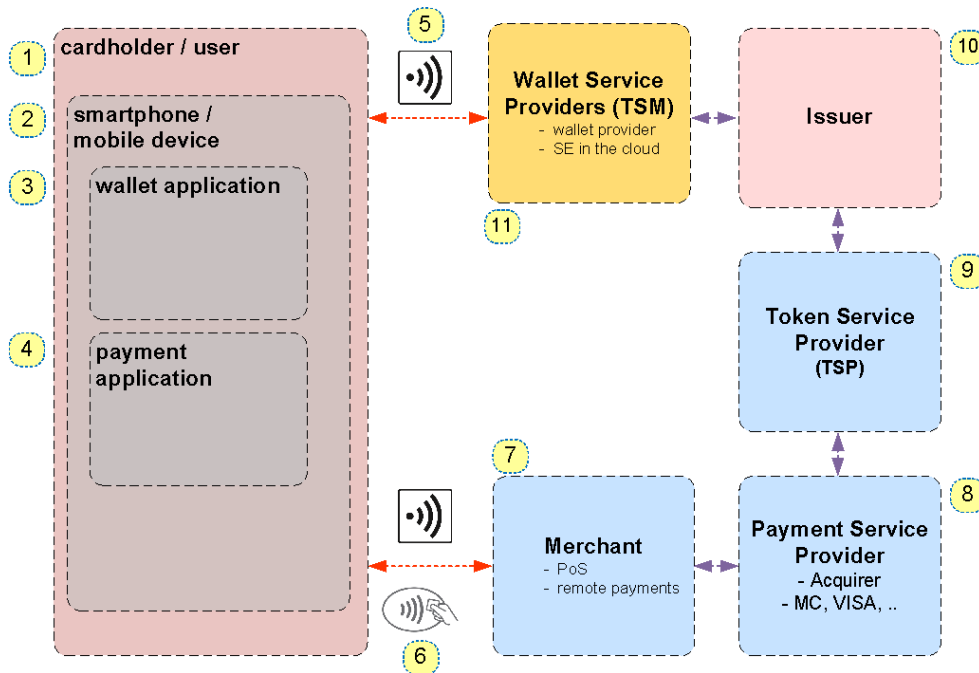


Figure 4:1 Overview of considered stakeholders in threat model

The following stakeholders represent the possible threat targets:

ID	Stakeholder	Comment
1	cardholder, user	The entity who initiates the mobile payment process as a customer and device owner.
2	smartphone, mobile device	The device used during the payment process.
3	wallet application	The wallet application that interfaces with the user.
4	payment application	The payment application, which can be part of the wallet or stored separately in a TEE or SE.
5	wireless interface	Wireless network interface used to connect to the service providers located in the cloud.
6	contactless interface	NFC connection between smartphone and NFC reader at PoS.
7	merchant	Merchant provides PoS contactless reader (NFC or MST) and terminal software.
8	payment service provider (PSP)	The PSP facilitates the connection into the payment network and offers various payment services for the merchant.
9	token service provider (TSP)	The TSP is part of the payment network and provides tokenisation services.
10	issuer	The bank that issues the cards and authorises them for tokenisation by the TSP.
11	wallet service provider (TSM)	Provides and manages the wallets, e.g., Samsung, Apple, or Google.

Table 4:1 Threat Targets

4.2 Evaluation –Threats to Mobile Payment Model

This chapter evaluates possible threats to selected targets. We employ STRIDE [65, 66], where applicable, to determine and categorise potential threats. At this stage, threats are also listed that are not specific to Apple Pay. This serves the purpose of being more general and applicable to all solutions and the ecosystem. In Chapter 5, Apple Pay is analysed in further detail regarding card enrolment and contactless card present (CP) transactions. The threats described below will be referenced accordingly.

Some threats are more relevant to certain processes. This is signalled in the ‘applies to’ column. These assignments will later be used to analyse weaknesses and strengths of the card enrolment and contactless payment (CP) processes.

Defines where it is detailed - card enrolment

Defines where it is detailed - PoS payment

Defines not specific to mobile payment and not further detailed

- **UA** Applies to security aware handling of payment device
- **SY** Applies to security posture of device, integration of the wallet into mobile OS
- **ER** Applies to enrolment process
- **CP** Applies to CP payment (contactless) process
- **CNP** Applies to card not present payment (remote payment) process

Threats to the cardholder:

ID	Threat to cardholder	STRIDE If applicable	Description	Applies To
CH1		S--I--	Phishing and social engineering to gain access to sensitive user information, which could be used for card enrolment.	UA
CH2		ST-I-E	Installation of malicious payment application from application stores, which will escalate local users' rights on the smartphone platform to gain sensitive information.	UA
CH3		-T-I-E	Installation of malware that tampers with the local file system and tries to access sensitive information.	UA
CH4		-T---E	Cardholder 'rooted' the smartphone and raises privilege level during normal operations. This makes the smartphone vulnerable to malware escalating the privilege level, and allows privileged access to the system.	UA

Table 4:2 Threats to the cardholder

Threats to the mobile phone / smartphone:

ID	Threat to smartphone	STRIDE If applicable	Description	Applies To
SP1		-T---E	Installation of malware that tampers with system services, including installation of rootkits to escalate privilege of user processes.	SY, ER, CP, CNP
SP2		-T-I-E	Unauthorised access to stolen or lost smartphone.	SY, CP, CNP
SP3		---I-E	Unauthorised information disclosure of cardholder data via NFC interface to third parties	SY, CP, CNP
SP4		---I-E	Unauthorised access to NFC interface and controller data to gain sensitive cardholder and payment information.	SY, CP, CNP
SP5		-T-I-E	Unauthorised access to smartphone data via offline methods, e.g., backup.	SY, CP, CNP
SP6		STR--E	Unauthorised access to identification and authorisation module to spoof cardholder during payment authorisation, including enrolment of adversary's fingerprints, and spoofing of original fingerprint.	SY, ER, CD, CNP
SP7		OWASP	Web application exploits during in-app payment or within standard purchasing portals	CNP
SP8		Platform Mgmt.	Lifecycle management of operating system software cannot be enforced on the device to provide necessary platform security.	SY, ER, CP, CNP
SP9		ST---D	Spoofing, tampering, or rendering the external services that the smartphone depends on unavailable, e.g., spoofing of domain name services or inserting rogue wireless access points.	SY, ER, CP, CNP
SP10		Platform Mgmt.	Threats to lifecycle management of payment device, e.g., lost, stolen, or replaced device.	SY, CP, ER, CNP

Table 4:3 Threats to the smartphone

Threats to the wallet application and payment application:

ID	Threat to wallet	STRIDE If applicable	Description	Applies To
SW1		-T---E	Installation of malware exploiting vulnerabilities of the wallet application.	SY, ER, CP, CNP
SW2		-T-I--	Reverse engineering of wallet and/or payment application to retrieve sensitive payment data.	SY, ER, CP, CNP
SW3		STRIE-E	Exploiting wallet vulnerabilities to gain unauthorised access to payment process. This includes software and authentication weaknesses.	SY, CP, CNP
SW4		--RI-E	Unauthorised access to confidential payment credentials belonging to wallet or payment application for fraudulent purposes.	SY, CP, CNP
SW5		S--I--	MITM attack between smartphone's network interface and external entities to gain access to confidential payment credentials.	SY, ER, CNP
SW6		---I-E	Unauthorized access to payment data on contactless NFC interface while in transition to PoS reader – general threat to NFC communication channel on 'air'.	CP
SW7		-TRI--	Installation of malicious wallet application from an application store, which will escalate local users' rights on the smartphone platform to gain sensitive information.	SY, ER, CP, CNP
SW8		-TRI--	Tampering with local payment application.	SY, ER, CP, CNP,
SW9		S-RI--	Disclosure of sensitive, original, not tokenised payment data used during enrolment, cross channel fraud.	SY, ER
SW10		Platform Mgmt.	Lifecycle management of payment application.	SY, ER, CP, CNP

Table 4:4 Threats to the wallet and payment application

Threats to the merchant:

ID	Threat to merchant	STRIDE If applicable	Description	Applies To
MT1		STRIE-E	Installation of PoS malware to gain access to payment credentials to conduct fraudulent payments, e.g., harvesting customer data for wallet enrolment ID&V process based on cardholder data such as CVV, cardholder name, expiry date, or purchase meta data.	CP
MT2		----D-	Render contactless reader unavailable due to DoS.	CP
MT3		---I--	MITM eavesdropping on contactless transaction channel for replying purposes.	CP
MT4		---I--	Unauthorised access to merchant's infrastructure.	CP, CNP
MT5		Platform Mgmt.	PoS software lifecycle management—see development kits (SDK) based on XP frameworks [75].	CP
MT6		OWASP [64]	Threats to the merchant's web application interface for remote payments.	CNP

Table 4:5 Threats to the merchant

Threats to the payment service provider:

ID	Threat to PSP	STRIDE If applicable	Description	Applies To
PS1		-TR---	Tampering with payment authorisation responses, rendering fraudulent transaction as valid, or changing data not enclosed by the message authentication code (MAC).	CP, CNP
PS2		S-RI--	MITM to access sensitive data for payment transaction flows into the payment network.	CP, CNP
PS3		-T-I-E	Gaining access to cardholder data stored on PSP infrastructure, e.g., CVV, cardholder name, expiry date, or other purchase meta data.	CP, CNP

Table 4:6 Threats to the payment service provider

Threats to the token service provider (TSP):

ID	Threat to TSP	STRIDE If applicable	Description	Applies To
TS1		-T-I-E	Unauthorised access to TSPs tokenisation services, CVV, cardholder name, expiry date purchase meta data, cryptographic key material, etc.	CP, CNP
TS2		-T-I-E	Reverse engineering of token mapping process, breach into the token lookup table.	CP, CNP
TS3		-T-I-E	Compromise/manipulation of anti-fraud measures, e.g., of domain restriction or allowed token-use window.	ER, CP, CNP
TS4		-T---D	Availability of TSP services (tokenisation, cryptogram validation, token lifecycle management, enrolment, HCE, domain restrictions, etc.)	ER, CP, CNP
TS5		S-R---	Re-use of payment cryptograms, use of tokenPAN for standard CNP.	CP, CNP
TS6		OWASP	Threats to the TSP's web services interface where other stakeholders need to make connections to.	ER, CP, CNP

Table 4:7 Threats to the token service provider (TSP)

Threats to issuer:

ID	Threat to Issuer	STRIDE If applicable	Description	Applies To
IS1		--R---	Fraudulent payment transactions.	CP, CNP
IS2		--R---	Payment transaction cryptogram–non-repudiation issues owing to derived shared keys (UDK).	CP, CNP
IS3		S-R---	Compromise of card enrolment services, enrolment of stolen cards, ID&V, yellow path.	ER
IS4		---I--	Integrity threats to meta data in ARQC not covered by MAC.	CP, CNP
IS5		S-R---	Compromise of payment authorisation process–reply attacks.	CP, CNP
IS6		---I-E	Compromise of cardholder data caused by data breach.	CP, CNP
IS7		---I--	Privacy threats to meta data provided by token requestor during enrolment.	ER
IS8		-T-I-D	Threat to lifecycle management of the cardholder's cryptographic keys (UDK). General key management aspect – not specific to mobile payment.	ER, CP, CNP
IS9		-----D	Availability of issuer services, such as payment authorisation (must be online), card enrolment, or ID&V.	ER, CP, CNP
IS10		Platform	Lifecycle management issuer's payment application, wallet application.	CP, CNP
IS11		S-R---	Fraud enforcement methods in case of compromised wallet or payment application.	CP, CNP

Table 4:8 Threats to the issuer

Threats to the wallet service providers (TSM):

ID	Threat to TSM	STRIDE If applicable	Description	Applies To
TM1		-----D	Availability of token requestor services (enrolment, token replenishment, token request, token lifecycle management).	ER, CP, CNP
TM2		---I--	Data breach of cardholder enrolment data (cardholder name, CVV, expiry date, PAN).	ER
TM3		---I--	Cardholder privacy issues owing to additionally collected meta data through TSM.	ER
TM4		Platform	Software lifecycle management of wallet or payment application.	CP, CNP

Table 4:9 Threats to the wallet service provider (TSM)

Chapter 5 Apply Threats, Evaluate Controls and Vulnerabilities

The security of mobile payments relies heavily on the strength of the authentication and registration controls (ID&V), considered within their individual mobile payment services. Therefore, on their side a mobile payment service provider should protect the initiation of mobile payments by protecting the card enrolment and access to sensitive payment enrolment data, using strong customer authentication], secure storage of payment credentials, and by protecting the communication channels. Similar protection methods have been proposed by ECB [62] in their recommendation for mobile payments. Specifically, weaknesses in the issuers ID&V process during card enrolment the enrolment process have led to the occurrence of ‘yellow path fraud’ [17], which in terms of fraud is an example of ‘account takeover’.

Note: Even though we were not allowed and entitled to conduct an intrusive vulnerability assessment, the executed threat analysis along with evaluation of the applied controls measures provides sufficient information to differentiate the three pay solutions.

5.1 Evaluate – Manual Enrolment Process

We will apply the evaluated threats to the card enrolment process and its involved stakeholders to find potential vulnerabilities. The card enrolment process, where an eligible credit card is manually added to the cardholder’s wallet, is the first step in the lifecycle of a digitised card. The details of the enrolment process are based on Apple’s documentation [26, 34], and partly verified by the results derived from the network analysis in Chapter 6.

The different steps and stakeholder involvement are summarised below, from top to bottom and from left to right. It shows how the stakeholders work together to achieve the required process assurance.

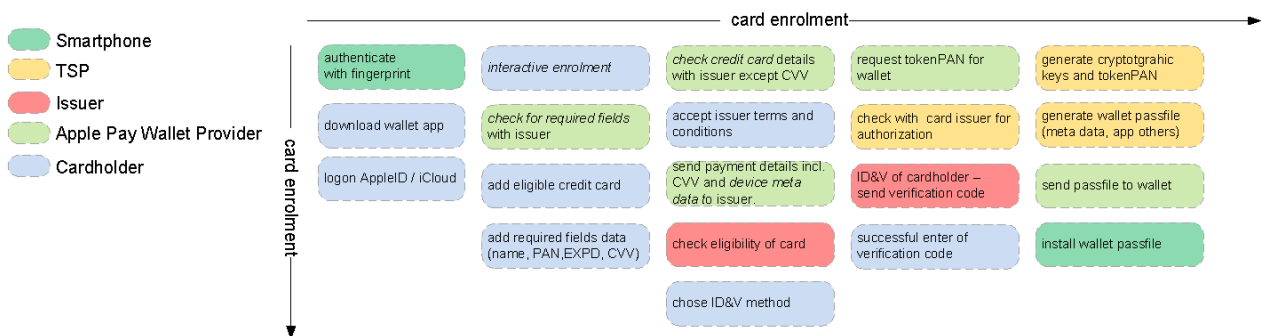


Figure 5:1 Apple Pay–Overview of Card Enrolment Steps

The enrolment process of the card schemes is mandated to be secured using at least two of the following step-up authentication methods as part of the EMV ID&V process [18]: call centres, one time passwords (OTP), or app-to-app authentication. See the image below, which is a visualisation derived from EMV tokenization specification [18].

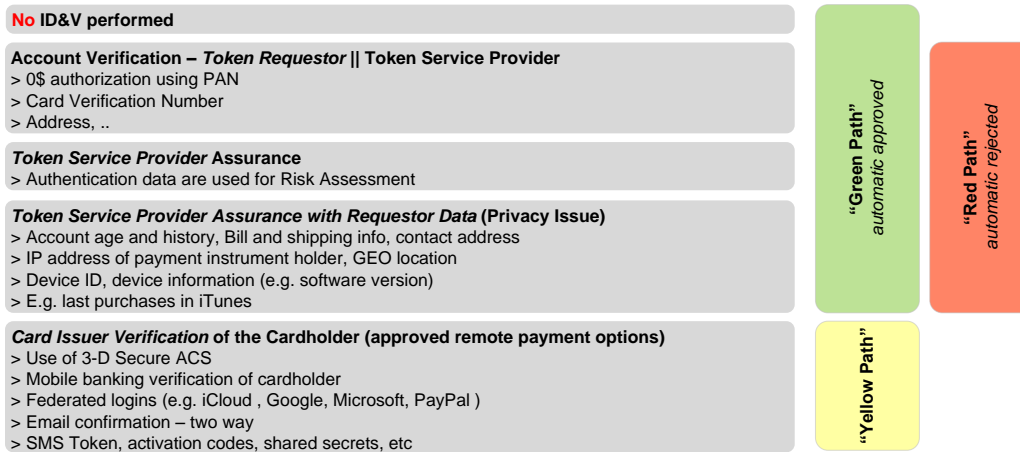


Figure 5:2 ID&V Methods

We see that ‘Yellow Path’ methods require an interaction on the cardholder side, whereas ‘Green Path’ methods can be automatically approved or rejected. Android Pay’s temporary charge of the cardholder account is one example. Using available meta data is the second one. All three wallet solutions comply with the initial requirement of using at least two step-up authentication methods.

Dataflow diagram-Apple Pay Enrolment:

The following data flow diagram has been derived from available apple documentation, in particular from the developer’s guide [26, 34]. It shows threats to the card enrolment environment. Only the threats assigned in Chapter 4.2 are displayed in the diagram.

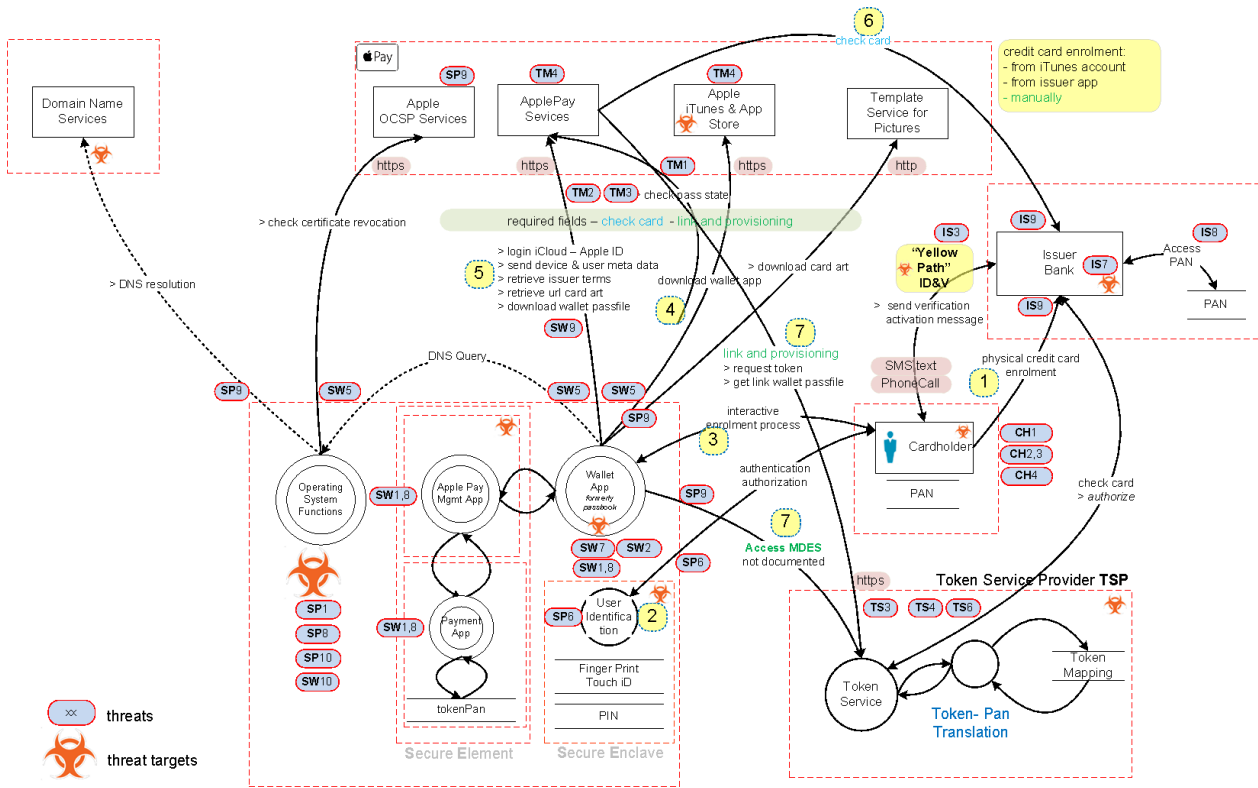


Figure 5:3 Apple Pay – Manual Card Enrolment

Note: Numbers (1-7) indicate the approximate flow through the ecosystem – please read Chapter 3.1 and review Chapter 6 for further information.

Focus on Cardholder: For the cardholder, we can identify the fundamental dilemma regarding information security, where a security-unaware user has specific security requirements but does not possess the necessary expertise. Improving user awareness with respect to the threats to mobile payments is essential. Threats, vulnerabilities, and control measures do not differ for the three eWallet solutions.

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
CH1	The cardholder is susceptible to phishing attacks, where attackers can collect confidential payment data that could be used for enrolment. The user needs to be made aware of necessary security procedures.			
	<i>control</i>	C_A1	C_A1	C_A1
CH2,3	The cardholder and device owner installs programs from an application store being unaware that they are malicious.			
	<i>control</i>	C_A2	C_A2	C_A2
CH4	The cardholder does root their device. The purpose does not matter in this case.			
	<i>control</i>	C_A2	C_A2	C_A2

Table 5:1 Threats to the Cardholder–Details

Control	Description
C_A1	Involved stakeholders must raise the cardholder’s awareness with respect to the safe handling of their pay-

	ment device and the possible consequences of liability shifts for fraudulent transactions resulting from negligent actions.
C_A2	Cardholders should consider treating their payment devices as securely as possible and protect them with malware protection tools.

Table 5:2 Threats to the Cardholder–Control Measures

Focus on Smartphones: The approaches to smartphone security are rather different for each of the three solutions. Apple’s is based on ‘security by design’, whereas Samsung Pay relies on the TMM KNOX software framework. Google’s Android is based on the idea the platform has already been compromised, and therefore it applies a layered security approach to mitigate potential risks. Even though malware attacks are primarily focused on Android, iOS attacks [69] are becoming more common. Recent attacks targeting all wallet operating system providers strengthen the idea that the principle of a layered security approach benefits all of them. There is no silver bullet to address all threats with one solution. Threat prevention, detection, and response methods should be available, and must work together to address threats. Furthermore, the platforms must be integrated into a patch and vulnerability management process to minimise the exposure to known exploits. Mobile device management (MDM) will play a key factor, even if this is done manually. Unfortunately, it is mostly enterprise users that benefit from expensive MDMs. Thus, security could be enforced by a third party (C_B5), by not allowing transactions based on device meta data.

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
SP1	Installation of rootkits that escalate privilege to access sensitive payment data, installation of key loggers, etc.			
	<i>control</i>	C_B1 C_B9	C_B2 C_B3 C_B9	C_B2 C_B3 C_B9
SP6	Tampering of authentication and authorisation module to authorise transactions.			
	<i>control</i>	C_B4	C_B4	C_B4
SP8	Software lifecycle management of smartphone system			
	<i>control</i>	C_B5	C_B5	C_B5
SP9	Spoofing, tampering, or rendering required external services unavailable.			
	<i>control</i>	C_B8	-	-
SP10	Threats to life cycle management of smartphone for the scenario of lost, stolen, or replacement phone.			
	<i>control</i>	C_B6 C_B7	C_B6	C_B6

Table 5:3 Threats to the Smartphone–Details

Control	Description
C_B1	Apple Pay secures sensitive payment data in its tamper-proof SE hardware, and does not allow other applications besides the wallet app to access the SE.
C_B2	Samsung Pay uses the TEE environment, whereas Android Pay applies whitebox-cryptography to protect sensitive payment data and programs. Both measures are based on software and its lifecycle management.
C_B3	To minimise data exposure, tokenisation is used together with dynamic key material. The keys are pre-fetched, only valid for a certain time, and restricted to the assigned token domain.
C_B4	Access to authentication module and fingerprint reader are protected via secure enclave on the iPhone and via trusted drivers and TEE on Android phones.
C_B5	Software lifecycle management is difficult to enforce. As a control measure for remote payments, I suggest using the 3-D Secure client SDK [81] as an additional measure and cancelling transactions outside of the allowed releases if not enforced by the wallet provider. See Chapter 2.6.
C_B6	All three solutions have various ways of disabling their smartphone and wallet applications. Because all three solutions apply tokenisation, only the token must be disabled at the side of the issuer or the TSP. No physical replacement is necessary.
C_B7	In the case that the iPhone is restored, marked as deleted, or set as lost in the iCloud, all tokenPANs and cryptographic key material will be deleted [34] from the secure element. A new enrolment must be initiated. This is also the case for a replaced iPhone. This is verified by the frequent connections to the iCloud. See network analysis in Chapter 6.
C_B8	External services must be protected for their availability and integrity, whereas the latter can only be checked on the endpoint–the payment instrument. The availability of most of Apple Pay’s services is protected by Akamai DDoS services. The integrity of SSL connection endpoints is possibly protected via certificate pinning : see OWASP [68] and passive network analysis in Chapter 6.3 for more details.
C_B9	Adversaries are increasingly targeting smartphones with malware. Therefore, it is advisable to install threat prevention technologies to address this risk [57].

Table 5:4 Threats to the Smartphone–Control Measures

Focus on Wallet Application and Payment Application: The wallet providers handle their application stores in different manners. Apple has a reputation for strict control of what is available via the app store. With this said, recent app store attacks [69] have shown that all three vendors are targeted with compromised applications. Such attacks cannot be completely prevented, but control measures must be able to detect such rouge applications or render compromised data useless for the attacker. The detection may be further enhanced via 3-D Secure device information [81], as soon as this is available.

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
SW1	Installation of malware exploiting vulnerabilities of the wallet application—see SP1.			
	<i>control</i>	see SP1		
SW2	Reverse engineering of wallet application.			
	<i>control</i>	C_C1	C_B2	C_B2
SW5	MiTM attack on wallet connections to the wallet provider and other external entities.			
	<i>control</i>	C_B8	-	-
SW7	Installation of malicious wallet application from the provider’s app store.			
	<i>control</i>	C_C2	C_C2	C_C2
SW8	Tampering with local payment application—see SP1.			
	<i>control</i>	see SP1		
SW9	Disclosure of sensitive, original, not tokenised payment data used during enrolment for cross channel fraud.			
	<i>control</i>	C_C3	C_C4	C_C5
SW10	Lifecycle management of payment application—see SP8.			
	<i>control</i>	see SP8		

Table 5:5 Threats to the Wallet and Payment Application—Details

Control	Description
C_C1	Apple Pay hosts the payment application in its tamper proof SE. Attempting to access the SE would render its data useless.
C_C2	All three app store providers provide code signing programs, which help to ensure that only legitimate software is downloaded from the app store. This does not prevent criminals from officially adding malware. App store providers need to further analyse their stored application for possible threats. The wallet provider must implement means to detect an integrity violation, which would allow the cancellation of enrolled credit cards.
C_C3	During card enrolment into the Apple Pay wallet, the cardholder enters their original payment details into the different panels, including the CVV code (see Chapter 0 (pointer 5,6,7) displaying the user view during enrolment). In the case that the smartphone is rooted, the payment information could be obtained via a key-logger, screen catcher, or similar technique. Such an attack is prevented by the security posture of the iPhone as a platform. After enrolment, Apple Pay does not store original sensitive payment information [31] in the cloud.
C_C4	First, the credit card is added to Samsung Pay using identical information as Apple Pay (C_C2). The card details are verified with the issuers. The vulnerabilities of data disclosure are the same as during the iPhone enrolment process, with the difference that the original payment details are stored in the Samsung cloud, which is a possible target for attacks.
C_C5	Users of Android Pay need to check-in their credit card with Google, who store the data in their Cloud. The cardholder ID&V is delegated to the issuer, but offers more options [39] than Samsung Pay, e.g., an additional temporary charge can be selected. The original payment data in the cloud is a possible target for attacks.

Table 5:6 Threats to the wallet application—Control Measures

Focus on Token Service Provider (TSP): The TSP hosts a wealth of sensitive payment data. Like the card issuers, they are regulated and assessed by PCI DSS [82, 13]. Here, a data breach would reveal a lot of cardholder payment data. Therefore, a TSP makes an attractive target for attackers, and must be well secured.

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
TS3	Compromise / manipulation of anti-fraud measures, e.g., of domain restriction or allowed token-use window.			
	<i>control</i>	C_D1	C_D1	C_D1
TS4	Availability of TSP services (tokenisation, cryptogram validation, token lifecycle management, enrolment, HCE, domain restrictions, etc.).			
	<i>control</i>	C_D2	C_D2	C_D2
TS6	Threats to the TSP's web services interface to which other stakeholders need to connect.			
	<i>control</i>	C_B8	C_B8	C_B8

Table 5:7 Threats to the TSP–Details

Control	Description
C_D1	In the case that an intruder can annul the fraud prevention measures that are provided by tokenisation, domain restrictions, and even time constraints on the use of dynamic keys, a fraudulent transaction could pass the first line of defence without detection. The required security measures are regulated by the card schemes and PCI DSS [82]. Protection levels have been steadily increased. Today, these should adhere to the security standards of NIST, SANS, and others.
C_D2	In case TSP services are not available, card enrolment will not be possible. Control measures are of a general character, and are the same as C_D1.

Table 5:8 Threats to the TSP–Control Measures

Focus on the Issuer: Issuers hosts a wealth of sensitive payment data. Issuers are regulated by PCI DSS [82] and other bodies. A data breach would probably reveal a lot of cardholder payment data, and is therefore an attractive target for attackers, and must be well secured. Specific and particularly interesting to mobile payment solutions are weaknesses in the ID&V [18] process during enrolment.

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
IS3	Compromise of card enrolment services.			
	<i>control</i>	C_E1	C_E1	C_E1
IS7	Privacy threats to meta data during enrolment.			
	<i>control</i>	C_E2	C_E2	C_E2
IS9	Availability of issuer services.			
	<i>control</i>	C_B8	C_B8	C_B8

Table 5:9 Threats to Issuer–Details

Control	Description
C_E1	The strength of ID&V methods is important to prevent the enrolment of stolen credit cards, and to ensure that the eligible cardholder is present. This is essential for all three solutions. There have been issues [8] where fraudsters could enrol stolen cards via 'yellow path' (Figure 5:2 ID&V Methods) enrolment.
C_E2	Apple Pay states in its privacy overview policy [31] that no sensitive payment data used during enrolment is stored. There was no detailed information available for the other wallet solutions regarding whether they store the original payment data or not. Regarding the content of other meta data, the wallet provider must comply with the corresponding data protection laws of the relevant jurisdiction.

Table 5:10 Threats to the Issuer–Control Measures

Focus on the Wallet Service Provider (TSM): The wallet service providers Apple, Samsung, and Google make interesting targets for attackers, as the wallets interoperate with them during a digitised card's lifecycle. The TSM integrity and availability are important for a functioning wallet app.

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
TM1	Availability of token requestor services			
	<i>control</i>	C_B8	C_F1	C_F1
TM2	Data breach of cardholder enrolment data			

		<i>control</i>	C_E2	C_E2	C_E2
TM3	Privacy issues regarding collected cardholder and meta data.				
		<i>control</i>	C_E2	C_E2	C_E2

Table 5:11 Threats to the Wallet Service Provider–Details

Control	Description
C_F1	Both of the other wallet solutions, based on HCE, need to have the TSM services online to frequently replenish their dynamic tokens. In the case of a depletion of their dynamic tokens, no payment is possible.

Table 5:12 Threats to the Wallet Service Provider-Control Measures

5.2 Evaluation–Contactless Payment at PoS (CP)

A contactless mobile payment solution must adhere to the physical contactless payment specification at the PoS reader. We will apply the evaluated threats to the contactless payment process and its corresponding stakeholders to determine potential vulnerabilities. The details of the payment process flow below are based on Apple’s documentation [26, 34] and the corresponding API descriptions. Practical tests have shown [Figure 10:5] that in other countries, where the contactless payment sign is present, the Apple Pay wallet works seamlessly. This was also apparent when the author performed tests with my card reader-supporting MasterCard *PayPass*. The contactless Apple Pay payment worked from the beginning [Figure 10:7].

Dataflow diagram - Apple Pay contactless payment:

Only the threats assigned in Chapter 4.2 are displayed in the diagram; others have been addressed in Chapter 5.1. The picture illustrates well that we have very little or no communication initiated by the smartphone; this seems logical, because on using the smartphone as a contactless payment card at PoS, it should also work without network connectivity, because Apple Pay wallet does not need frequent replenishment of its cryptographic keys.

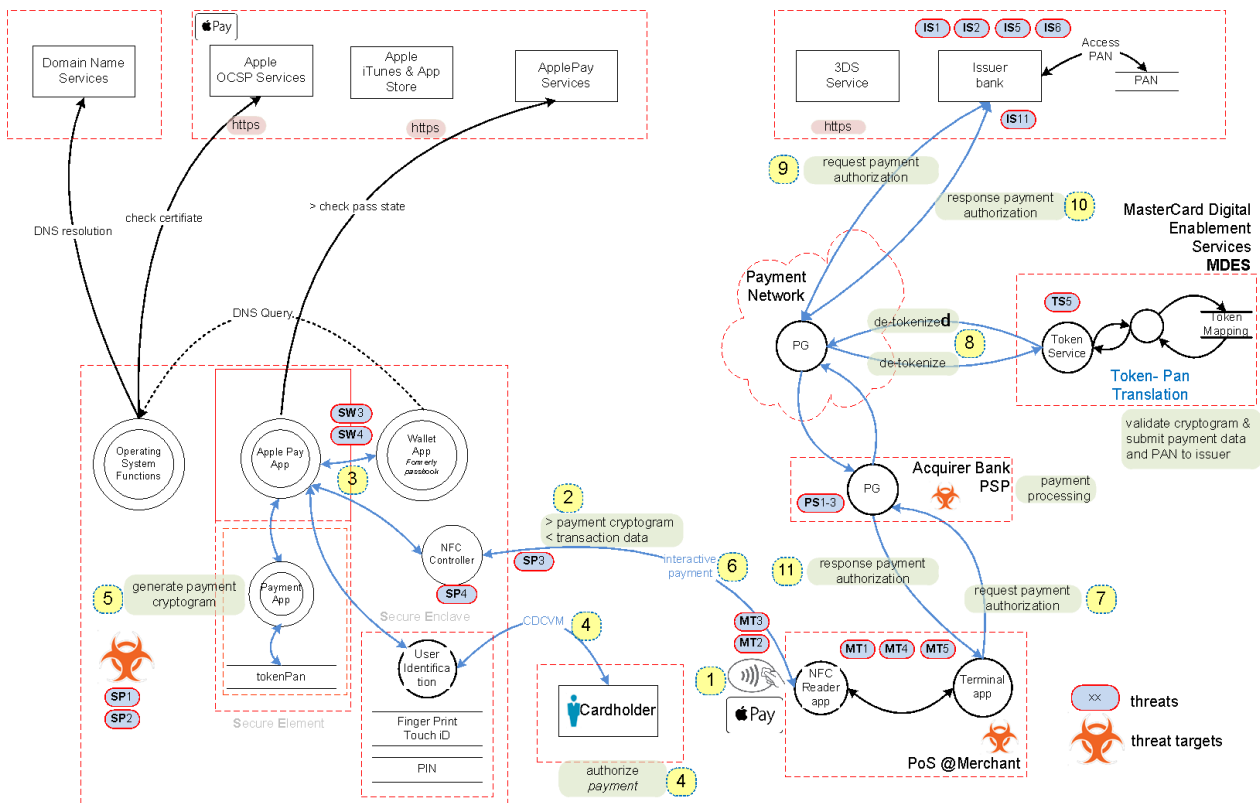


Figure 5:4 Apple Pay–Contactless Payment PoS

Note: The numbers (1-11) indicate the approximate flow through the ecosystem.

Focus on Smartphone:

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
SP2	Unauthorised access to stolen or lost smartphone			
	<i>control</i>	C_M1	C_M1	C_M1
SP3	Unauthorised information disclosure of cardholder data via NFC interface to third parties			
	<i>control</i>	C_M2	C_M2	C_M2
SP4	Unauthorised access to NFC interface and controller data to gain sensitive cardholder and payment information.			
	<i>control</i>	C_M3	C_M3	C_M3

Table 5:13 Threats on Smartphone–Details

Control	Description
C_M1	In the case that an iPhone is stolen or lost, the wallet application can be remotely disabled via [31] reporting the lost phone or using the iCloud settings. The TSP or issuer will suspend the payment device, regardless of whether it is online. Samsung Pay [47] has a similar method. For all solutions, the card issuer can disable the payment device at any time.
C_M2	Access to contactless card data is only possible in the case that the cardholder approves this via pin or fingerprint. This in contrast with contactless cards, for which sensitive payment data can be retrieved without authentication. See the screenshots in Figure 10:4. The disclosure of the tokenPAN (device account number) does not open the door for fraud, as no access is possible to the keys used to generate the cryptograms.
C_M3	Apple Pay's NFC interface is not open to third parties [34], in contrast to the Android platform. This strict control of the involved payment components significantly reduces the attack surface.

Table 5:14 Threats to the Smartphone–Control Measures

Focus on Wallet Application and Payment Application:

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
SW3	Exploiting wallet vulnerabilities to gain unauthorised access to payment process.			
	<i>control</i>	C_N1	C_N1	C_N1
SW4	Unauthorised access to confidential payment credentials belonging to wallet or payment application			
	<i>control</i>	C_N2	C_N2	C_N2

Table 5:15 Threats to the Wallet Application–Details

Control	Description
C_N1	For Apple Pay, all of the involved wallet components are under strict control, and others are not allowed to access them. In general, Apple manages its software reasonably well, and because the payment application is strictly separated and access is restricted to Apple only, there is no third-party software to be considered during updates. This in contrast with Android-based solutions, where the operating system version can vary significantly, and more components are involved during an update. Hence, the quality of the security also varies. Lifecycle management of the wallet application and operating system is essential. See SW-10, software lifecycle management.
C_N2	All three solutions only store the tokenPAN and static or dynamic keys. Apple stores its sensitive material in a tamper proof SE. The other solutions work with dynamic key material via HCE. There, the data is secured within the TEE, the cloud SE, or via whitebox cryptography.

Table 5:16 Threats to the Wallet Application–Control Measures

Focus on Merchants: The PoS at the merchant site remains an important attack target, as shown by the fraud figures from FRAUD Action [56]. However, focusing on mobile payment data that is vulnerable at the merchant site, the use of tokenisation eliminates possible cross channel fraud, as the tokenPAN cannot be used without a valid cryptogram (see Chapter 2.5). Our own tests using the tokenPAN in a CNP payment transaction revealed a failed transaction, where the payment service provider cancelled the transaction (see Chapter 10.3.5). Therefore, the possible impact on a merchant using tokenisation and its mobile payment data can be neglected.

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
MT1 MT5	The installation of PoS malware to gain access to payment data using vulnerabilities of the installed software.			

		<i>control</i>	C_O1	C_O1	C_O1
MT2	Render contactless reader unavailable due to DoS. This attack is difficult to prevent.				
		<i>control</i>	-	-	-
MT3	As described, MITM eavesdropping on contactless for transaction channel for replying purposes remains possible [89].				
		<i>control</i>	C_O2	C_O2	C_O2

Table 5:17 Threats to the Merchant–Details

Control	Description
C_O1	PoS reader software is a very attractive attack target, and in the case, that the software is outdated an attack is even easier. Vulnerability and patch management is an important factor for this environment. An example is MICROS' data breach [58]. The use of payment tokens minimises the fraud risk. <i>Note: Even the SDK for ACS that we purchased is designed for WinXP [75].</i>
C_O2	As mentioned before, retrieved mobile payment data cannot be used to reply, owing to unique payment cryptograms (see Chapter 2.4.5). Therefore, known replay attacks [59] in contactless payment scenarios are not applicable. General NFC vulnerabilities are still present, but not of much use for fraudsters.

Table 5:18 Threats to the Merchant–Control Measures

Focus on Payment Service Provider: Mobile payment does not introduce new threats or vulnerabilities into the PSP environment, and therefore is not further analysed. On the contrary, owing to the use of tokenisation and dynamic cryptograms, highly regarded PAN data is mostly useless for fraudsters.

Focus on Token Service Provider: Threats and vulnerabilities to the TSP infrastructure are not specific to mobile payment solutions, and are not further detailed. Some security recommendations are included in the EMV Tokenization Specification [18], and in PCI's recommendations [9, 10, 13].

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
TS5	Re-use of payment cryptograms and use of tokenPAN for standard CNP.			
	<i>control</i>	C_P1	C_P1	C_P1

Table 5:19 Threats to the TSP–Details

Control	Description
C_P1	TokenPAN ranges must not overlap with ordinary credit card PAN ranges, in order to strictly separate them. This is stated in EMV's Tokenization Specification Chapter 8.2 [18]. The TSP will reject a PAN without a matching cryptogram. Our own test has also demonstrated this behaviour (see Chapter 10.3.5).

Table 5:20 Threats to the TSP–Control Measures

Focus on Issuers: Issuers must keep the fraud figures down. The introduction of mobile payments for CP contactless and CNP will soon play an important role—see also Chapter 7 on Apple Pay and DSRP—How to Improve CNP.

Threat ID	Description	Apple Pay	Sams Pay	Andr Pay
IS1	Fraudulent payment transactions.			
	<i>control</i>	C_Q1	C_Q1	C_Q1
IS2	Payment transaction cryptogram–non-repudiation issues owing to shared keys (UDK).			
	<i>control</i>	C_Q2	C_Q2	C_Q2
IS5	Compromise of payment authorisation process–reply attacks, see TS5.			
	<i>control</i>	See TS5		
IS6	Compromise of cardholder data—in the case of a data breach.			
	<i>control</i>	C_Q3	C_Q3	C_Q3
IS11	Fraud enforcement methods in the case of a compromised wallet or payment application.			
	<i>control</i>	C_Q4	C_Q4	C_Q4

Table 5:21 Threats to the Issuer–Details

Control	Description
C_Q1	Mobile payment devices offer additional meta data compared with conventional contactless or contact cards. This opens the door for advanced fraud analytics. See Chapter 2.6 on the Role of Meta Data in Fraud Prevention and 3-D Secure 2.0.0.

C_Q2	The use of a shared key is not a specific threat for mobile payment. Rather, this is a design issue, and therefore is not further detailed.
C_Q3	The issuer works with the real PAN. Therefore, a data breach is not specific to mobile payments, and is not further analysed. Tokenisation for the PAN data at rest is recommended by the PCI Standards Council [9, 10], but is not mandatory.
C_Q4	Additional meta data can help to detect and prevent fraudulent transactions. See C_Q1.

Table 5:22 Threats to the Issuer–Control Measures

5.3 Interpretation and Conclusion

Though the three wallet solutions implement the mobile payment solution and its security differently, they all are fully compliant to the EMV contactless [85, 86, 87] payment specification and EMV tokenization [18], and are supported by the card schemes. They allow using a smartphone as a contactless credit card (near field communication device) at a point of sales (PoS) and are compatible to MasterCard’s *PayPass* and Visa’s *payWave* specification.

The main message of the threat analysis is that the application of EMV tokenization and the EMV-based payment cryptograms help to reduce the impact of compromised sensitive cardholder information on CNP cross-channel fraud figures and that they do so irrespective of whether the solution uses SE, TEE, or HCE technology to protect the sensitive payment data. Due to the compatibility requirements on contactless payments at a PoS, known vulnerabilities of the NFC transport channel are still present, but tokenization will limit a fraudster’s ability to leverage the stolen payment data.

The derived data flow diagrams illustrate how the wallet communicates, pinpoints the possible attack targets and demonstrates the widening of the attack surface caused by the introduction of new stakeholders such as TSP and wallet providers. Where the new services are offered via the Internet, the payment network applies approved security controls (See Figure 5:5). Apple Pay’s passive network analysis reveals that it addressed the threats well. Because the mobile payment applications blend in with the existing payment network, those existing components inherently possess the same vulnerabilities as before except for the tokenized payment data. However, how well the new payment solutions perform in terms of security cannot be evaluated without conducting an intrusive vulnerability assessment of the ecosystem and its stakeholders.

Conclusion & Recommendation: As proposed by Dieter Gollmann [90], the 1st fundamental design decision asks for ‘where to focus security controls’. To build a secure mobile payment ecosystem, the control components and stakeholders must work together in a balanced *prevent*, *detect* and *response* mode to address the threats (See NIST Cyber Security Framework [67]).

The figure below shows how different control measures (top) work together to achieve the required security posture at different points of the mobile payment ecosystem.

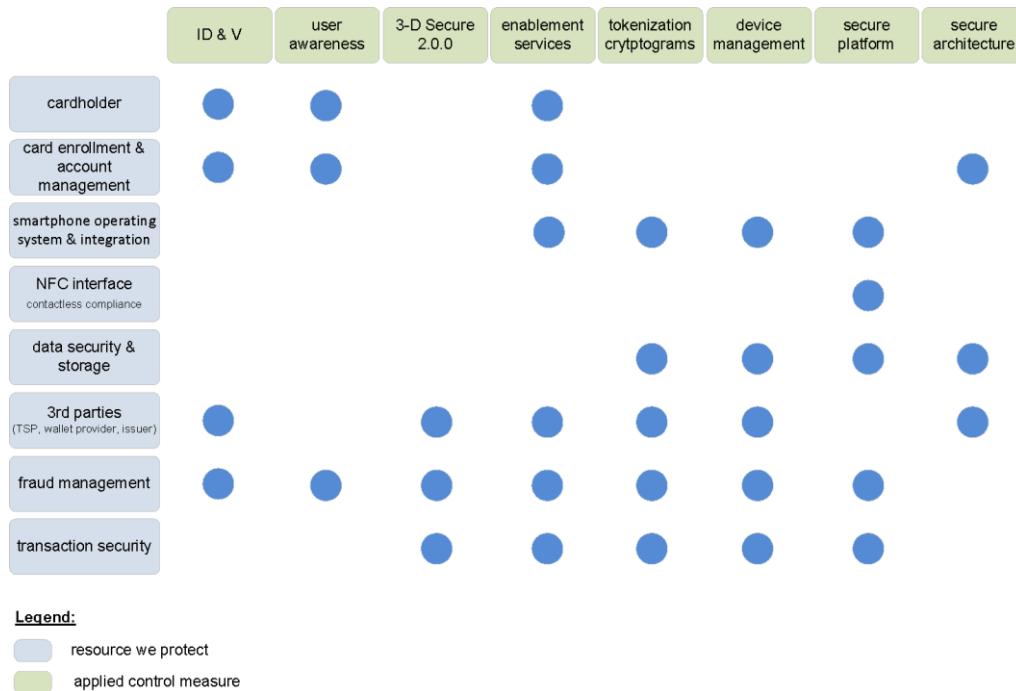


Figure 5:5 Mobile Payment–Layered Security–Attack Targets and Controls

Apple’s ‘Yellow Path’ [8] initial issuer ID&V weakness showed how layered security fails if one component fails. It led to the enrolment of stolen credit cards irrespective of the security of the iPhone operating system. An analogue example related to tokenization would be if the TSP does not enforce domain restriction and replay protection of the payment cryptograms. The different control measures need to work together.

We recommend focussing on the following targets and controls irrespective of the ‘Pay’ solution:

Cardholder: The dilemma is that a security unaware user must follow specific security requirements. The cardholder, i.e. the owner of the payment device, needs to *be made aware* of the security risks involved with using this novel payment option. He or she needs to be educated about the risk factors such as careless use (e.g. rooting), exposing the device to internet born threats like malware infections, losing the device or choosing a weak password. User awareness is essential for every security strategy.

Card enrolment: Weaknesses in the card enrolment process can lead to the enrolment of stolen credit cards. The strength and the combination of the applied identification and authentication methods (ID&V) must provide the necessary assurance. We recommend leveraging a smartphone’s additional capabilities (e.g. fingerprint, face recognition, OTP) and the metadata it provides to address fraudulent attempts.

Smartphone operating system & integration: Besides Apple Pay, which uses a dedicated SE and NFC controller to protect the sensitive payment process, the smartphones were not designed with payment security in mind. The software lifecycle management of smartphones is a critical security factor. For instance, how do you entice users to upgrade their Samsung smartphone when there is no reason besides re-establishing the Samsung Pay security level? Apple’s Secure Element is a dedicated software module, and thus an update can be conducted instantly and without many dependencies on other software modules. From my point of view, this reduces the complexity and improves the security. Compensating measures for the other ‘Pay’ solutions are to use HCE in combination with tokenization and dynamic key material, which needs frequent replenishment. Another option is to employ policy enforcement by the wallet provider or by the TSP based on available metadata such as 3-D Secure Device Information [81]. Advanced analytics will help to detect anomalies and enable the stakeholders to act upon such anomalies.

NFC Interface: With the exemption of Apple Pay, The NFC interface can not only be used by the wallet application, but also by other applications. For instance, malicious NFC tags could be used to initiate an attack. The attack surface is enlarged, and the security posture weakened. User awareness, platform hardening and installation of threat prevention software [57], applied tokenization, and dynamic cryptograms count as compensation and prevention measures. The vulnerabilities related to confidentiality and service availability (DoS) with respect to the NFC communication channel on air are still present and not further addressed in this paper.

Data security and storage: As described in the smartphone integration, the endpoint's platform security is critical for deciding which data can be stored where. Android addresses the threat of a compromised operating system to the payment process by using HCE technology, Samsung uses a similar approach, while Apple Pay with its integrated smartcard (SE) protects the payment data with local security measures and relinquishes the HCE approach. During enrolment into the wallets, we have sensitive payment data at rest and in motion. All three wallet solutions ask for the original credit card details during enrolment, which is not protected by tokenization at this point. The security of this data depends on the security of the wallet provider when storing those details in their cloud and the security posture of the smartphone payment device. However, companies processing payment data must comply with PCI DSS [82].

3rd parties – TSP, wallet providers: The new stakeholders of the payment ecosystem logically increase the present attack surface. The services offered via the Internet and accessible via web services are well-protected using established security controls to address DDoS, confidentiality, or non-repudiation such as Apple Pay's digital secure remote payment (DSRP). Those new stakeholders are likely targets of attacks and must be protected with an adaptable security architecture. Targeted attacks are present in these environments such as the breach of Micro's point of sales division [58] or recent attacks on the SWIFT's payment network and software [10-7].

Fraud management: Fraud management and its sensors, i.e. metadata, play an essential role and must adapt to the future development of the threat landscape. The attack vectors for fraud are manifold. To come back to the cyber resilience of the payment ecosystems, we must address prevention, detection, and response mode. With the integration of domain restriction, tokenization and the additional metadata for advanced fraud analytics, the stakeholders are more likely to detect fraudulent transaction attempts at different places and respond to it. Because the digitized credit card in the wallet is not a physical credit card, the issuer's options to intervene with the lifecycle are becoming very efficient by the instant use of the web interface to replace, withdraw or re-issue a new tokenPAN.

Transaction security: The transaction security benefits from all domains mentioned before. However, tokenization and EMV payment cryptograms build the foundation of the mobile payment ecosystem

Chapter 6 Network Analysis - Card Enrolment Apple Pay

6.1 Scope of Network Analysis

We do not carry out an active vulnerability assessment concerning Apple Pay's service infrastructure, as this would contravene CMA 1990 (Computer Misuse Act) [78]. These restrictions also apply to any reverse engineering attempts of software or jail breaking of the iPhone, because such acts could constitute an infringement of intellectual property rights. Regarding iPhone wallet connections into the Apple Cloud, these are encrypted and pinned to a certificate chain, and we will not be able to observe the encrypted traffic for further protocol analysis. However, we can detect the communication flows involved during different user actions. Overall, we will not violate any legal boundaries during this network analysis.

6.2 Setup Description

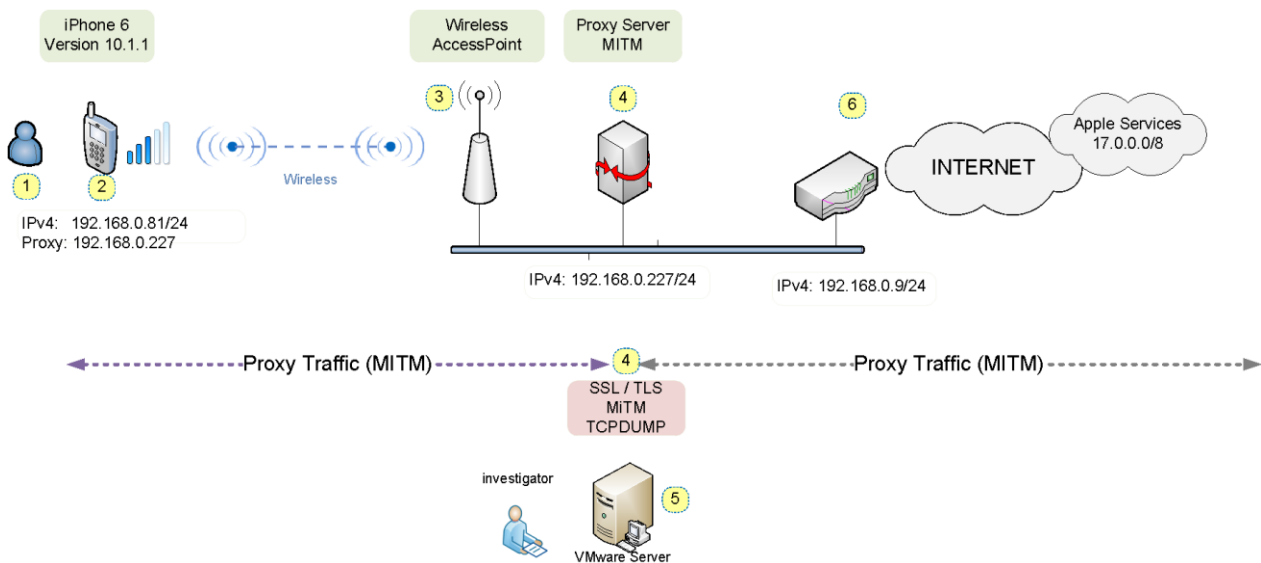


Figure 6:1 Network Analysis–Apple Pay Card Enrolment

Test Setup Description:

Please refer to Chapter 0 for further information regarding the devices and software components used.

ID	Description
1	User enrolling their credit card into the wallet.
2	iPhone 6 is manually configured with IP address of DNS and proxy. We also added the SSL root certificate of the proxy server to the profile section to declare it as trusted, in case we perform SSL/TLS interception on the proxy server. Communication via 3G/4G has been disabled, to force all traffic through the proxy server.
3	Wireless access point is used for internet connectivity.
4	Proxy server acting as approved MiTM and used to log all access. Proxy server runs as a virtual machine on a MacOSx host, where we also take the network dump.
5, 6	The wireless router provides access to the internet.

Table 6:1 Setup Network Analysis

6.3 Enrolment Process - with SSL/TLS Interception-‘Failed’

The Apple Pay wallet application, including other services such as iCloud and iTunes, does not allow a ‘man in the middle’ to intercept SSL traffic. Other encrypted sessions run via the standard browser Safari work effectively. Native Apple iOS applications verify the keychain during a TLS negotiation, and compare this with the trusted one available on Apple support [30]. In addition, Apple’s developer guide mandates the clients to evaluate the trust of the server [29], which must have failed even though the root signing certificate (3) of the MiTM web proxy has been added to the local trust store. Additional information is included in the iOS security guide [34] under ‘App Transport Security’.

Conclusion: The writer has expected Apple to use certificate pinning on its encrypted TLS communication channels to prevent MiTM attacks to protect the confidentiality of the transport channel. Thus, the wallet app does not permit to connect to the Apple Pay services and disconnects the session.

Evidence:

- (1) Error message ‘Could Not Connect to Apple Pay’ on the iPhone while starting the card enrolment process.
- (2) Packet trace session is disconnected from the client, probably after validating the certificate chain for its trust.
 - (2a) TCP/IP session is established (packet 334).
 - (2b) iPhone sends connection [RST] and the session terminates (packet 335).
- (3) MiTM certificate added to the profile store.

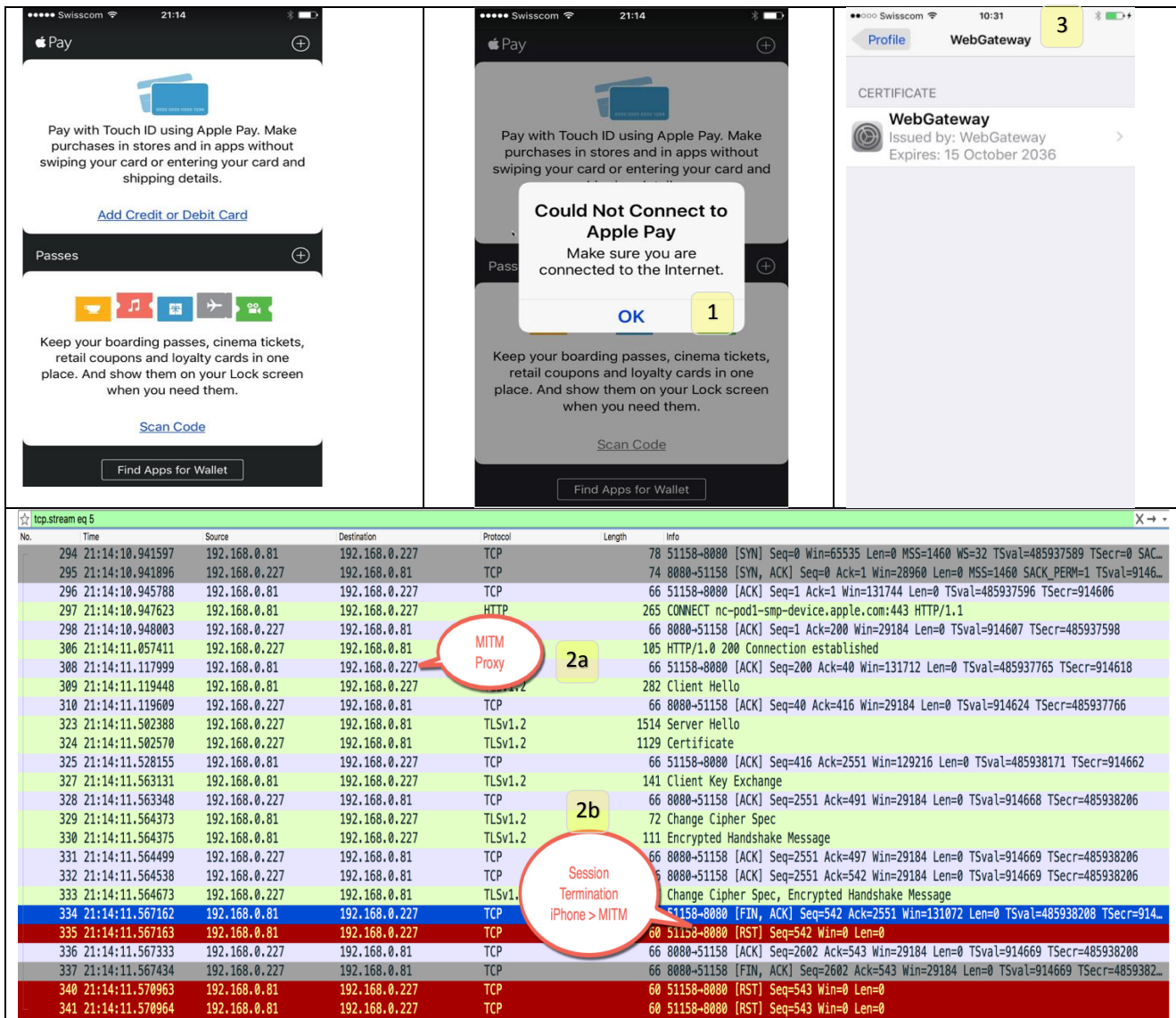


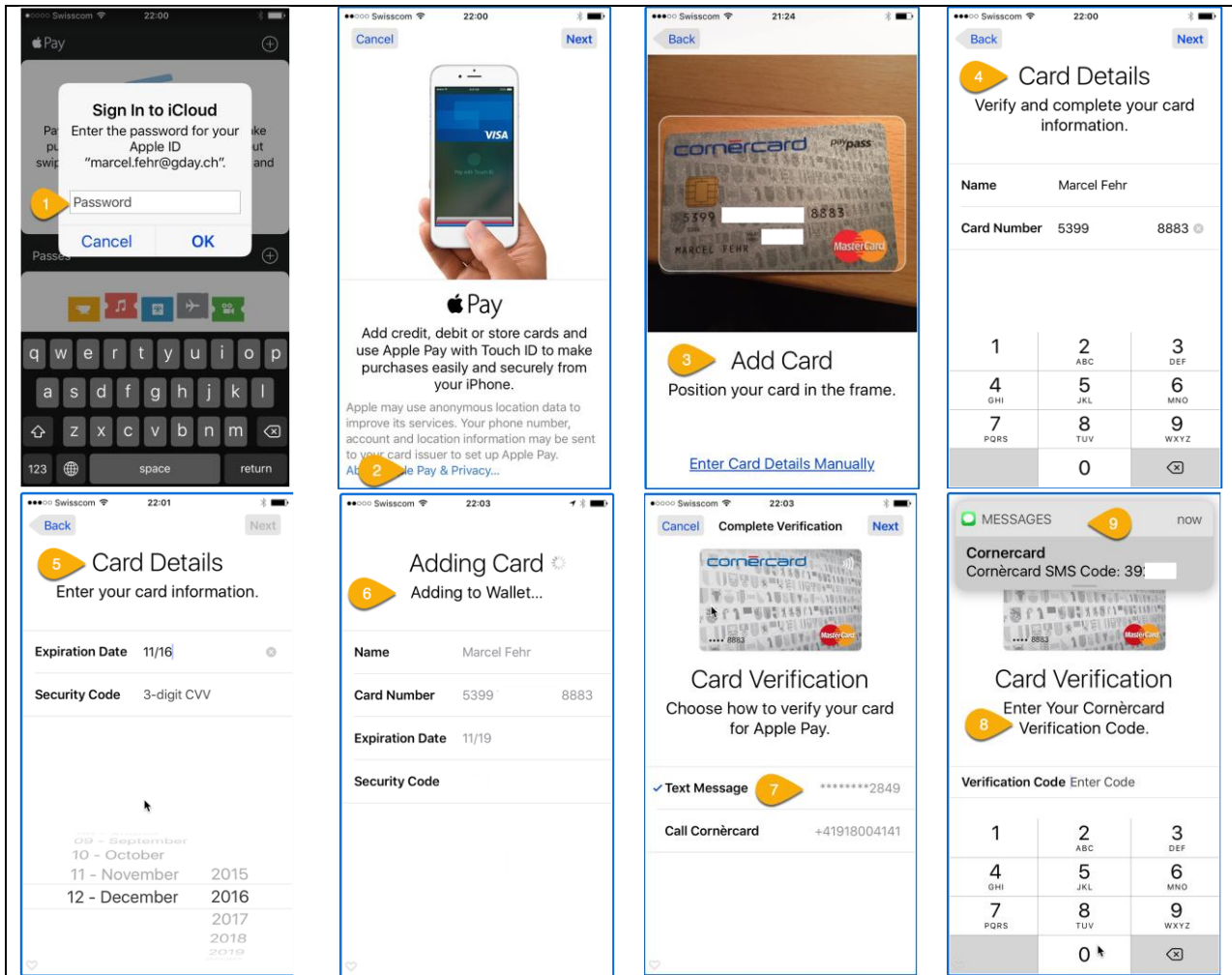
Figure 6:2 Network Analysis, Apple Pay Card Enrolment, Failed with SSL Interception

6.4 Enrolment Process without SSL/TLS Interception-‘Success’

The MiTM web proxy has been configured *not* to intercept SSL/TLS traffic. Even though we will not be able to analyse what is inside the encrypted connection, we still can detect the destinations with which the iPhone is communicating during the enrolment process.

6.4.1 Enrolment Process – User View

The images below show an example enrolment process for the chosen issuer. This maps to the process displayed in Figure 5:1, from top to bottom and then left to right. The numbers indicate the process sequence.



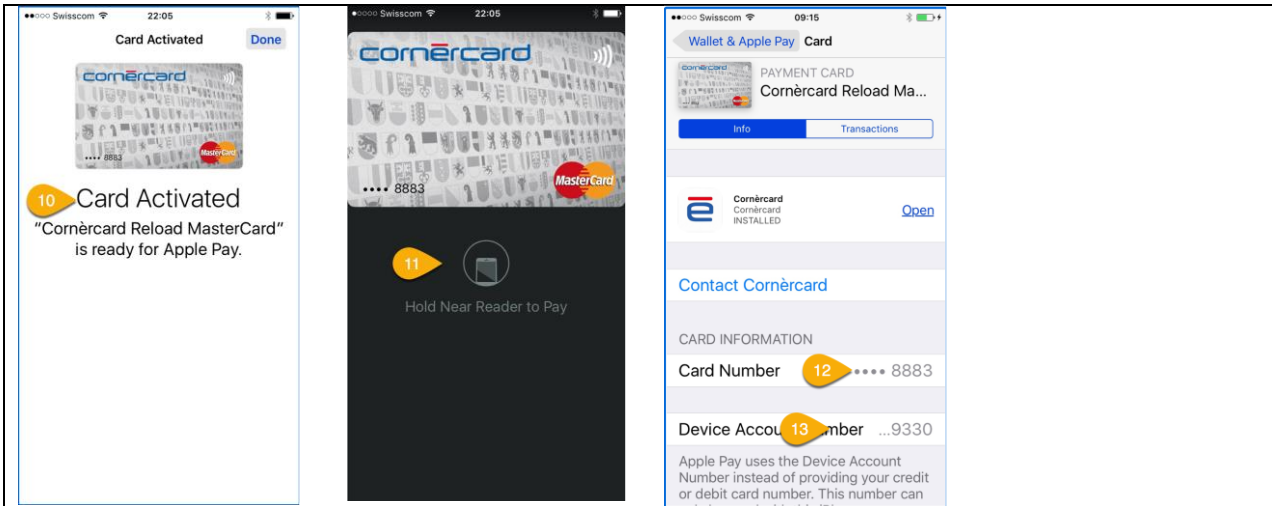


Figure 6:3 Network Analysis of Apple Pay Card Enrolment-Success

6.4.2 Enrolment Process–Network View

We have taken a network *tcpdump* and a detailed http trace on the web proxy. Because we do not have insight into the encrypted transport channel, we do not further consider the *tcpdump* output.

Http trace on web proxy:

Please refer to the trace output in Figure 10:1. The table below shows the various destinations. At the same time, we extracted the http: *user-agent* header from the connection. The name may be an indicator of which iPhone component could have initiated the connection.

ID	Access to	User-Agent	Interesting
1.	gsa.apple.com	Wallet	
2.	configuration.apple.com	geod	
3.	gsas.apple.com	akd	
4.	keyvalueservice.icloud.com	SyncedDefaults	
5.	p23-keyvalueservice.com	SyncedDefaults	
6.	gsa.apple.com	akd	
7.	gsas.apple.com	akd	
8.	nc-pod1-smp-device-asset.apple.com	Wallet	
9.	ocsp.apple.com/ocsp04-applesica301/...(http)	security	cert validation
10.	ocsp.apple.com/ocsp04-appleroocag3/...(http)	security	cert validation
11.	nc-pod1-smp-device.apple.com	Wallet	
12.	pr-pod1-smp-device.apple.com	Wallet	
13.	nc-pod1-smp-device.apple.com	Wallet	
14.	nc-pod1-smp-device.apple.com	seld	secure element
15.	nc-pod1-smp-device.apple.com	Wallet	
16.	pr-pod1-smp-device.apple.com	Wallet	
17.	nc-pod1-smp-device.apple.com	Wallet	
18.	ocsp.apple.com/ocsp03-wwdr02/...(http)	security	cert validation
19.	nc-pod1-smp-device.apple.com	passd	
20.	init.itunes.apple.com	location	
21.	play.itunes.apple.com	location	
22.	xp.apple.com	itunesstored	
23.	gsp10-ssl.apple.com	location	
24.	p23-fmfmobile.icloud.com	FMFD	
25.	configuration.apple.com	geod	
26.	p23-keyvalueservice.com	SyncedDefaults	
27.	pr-pod1-smp-device.apple.com	Wallet	
28.	p23-keyvalueservice.com	SyncedDefaults	
29.	nc-pod1-smp-device.apple.com	seld	secure element

ID	Access to	User-Agent	Interesting
30.	init.itunes.apple.com	locationd	
31.	play.itunes.apple.com	locationd	
32.	xp.apple.com	itunesstored	
33.	p23-fmfmobile.icloud.com	FMFD	
34.	nc-pod1-smp-device-asset.apple.com	Wallet	
35.	nc-pod1-smp-device.apple.com	Wallet	
36.	nc-pod1-smp-device.apple.com	passd	
37.	tds.mdes.mastercard.com	passd	TSP
38.	nc-pod1-smp-device.apple.com	passd	
39.	init.itunes.apple.com	com.apple.Passbook	
40.	play.itunes.apple.com	itunesstored	
41.	xp.apple.com	itunesstored	
42.	sp.itunes.apple.com	com.apple.Passbook	
43.	nc-pod1-smp-device.apple.com	Wallet	
44.	tds.mdes.mastercard.com	passd	TSP
45.	init.itunes.apple.com	com.apple.Passbook	
46.	play.itunes.apple.com	itunesstored	
47.	xp.apple.com	itunesstored	
48.	a1.mzstatic.com/eu/r30/xxx (http)	Wallet	Issuer Picture

Table 6:2 Network Analysis-Connections

6.5 Analysis-Network View

Based on the web proxy trace, further investigate how much useful information can be found from the connection trace. Analysing the network view provides a picture of which components are exposed to the internet, thus belonging to the attack surface during card enrolment. We are not allowed to conduct a typical network vulnerability assessment, where we would use the KALI pentest suite [77] with its plethora of tools. Using these would be not distinguishable from an ordinary attack, and would contravene CMA 1990 [78].

Certificate Verification: The iPhone regularly verifies the validity of certificates used via the Online Certificate Status Protocol (OCSP). See packet id_9, 10, and 18 in Table 6:2.

Server-Side Connections: According to the Apple iOS security guide [34], ‘Apple Pay uses three server-side calls to send and receive communication with the card issuer or network as part of the card provisioning process: Required Fields, Check Card, and Link and Provision. The card issuer or network uses these calls to verify, approve, and add cards to Apple Pay.’ These three server-side calls cannot be identified in our trace. We see two other server-side calls to the token service provider (MDES), see packet id_37 and id_44 in Table 6:2.

Service Hosting/DDoS Protection: We analysed the DNS names. See Figure 10:1 Network Analysis– for more details. The DNS analysis shows that Apple runs a large part its customer facing environment behind Akamai content delivery networks, which also provide DDoS protection. This can be inferred because the address records of most services point towards an AKAMAI cname, which in turn point towards an Apple owned IP address or an AKAMAI owned IP address.

The output shown below can be retrieved using the Linux command #dig ‘DNS NAME’. See Figure 10:2 for more details.

DNS Name	Resolves to CNAME or Address
a1.mzstatic.com	a1.mzstatic.itunes-apple.com.akadns.net. a1.mzstatic.com.edgesuite.net..
configuration.apple.com	configuration.apple.com.edgekey.net. 5153.e9.akamaiedge.net.
gsa.apple.com	gsa.apple.com.akadns.net. 17.171.74.166
gsas.apple.com	gsas.apple.com.akadns.net. 17.141.5.97
gsp10-ssl-apple.com	sp10-ssl.ls-apple.com.akadns.net. 17.167.193.162
init.itunes.apple.com	init-cdn.itunes-apple.com.akadns.net. itunes.apple.com.edgekey.net.
keyvalueservice.icloud.com	keyvalueservice.fe.apple-dns.net. 7.248.146.110
nc-pod1-smp-device-asset.apple.com	e9959.e9.akamaiedge.net. nc-pod1-smp-device.gcsis-apple.com.akadns.net.
nc-pod1-smp-device.apple.com	c-pod1-smp-device.gcsis-apple.com.akadns.net. 17.171.78.6
ocsp.apple.com	ocsp.pki-apple.com.akadns.net. 17.171.8.16
p23-fmfmobile.icloud.com	p23-fmfmobile-current.edge.icloud.apple-dns.net. 17.248.146.181
p23-keyvalueservice.icloud.com	p23-keyvalueservice-current.edge.icloud.apple-dns.net. 17.248.146.175
play.itunes.apple.com	play-cdn.itunes-apple.com.akadns.net. itunes.apple.com.edgekey.net.
pr-pod1-smp-device.apple.com	pr-pod1-smp-device.gcsis-apple.com.akadns.net. 17.141.128.6
sp.itunes.apple.com	sp-cdn.itunes-apple.com.akadns.net. sp.itunes-apple.com.akadns.net.
tds.mdes.mastercard.com	216.119.218.153
xp.apple.com	xp.itunes-apple.com.akadns.net. mt-ingestion-service-mr22.itunes.apple.com.

Table 6:3 Network Analysis–DNS Resolution Overview

Location Analysis: As Apple Pay uses privacy-sensitive information during enrolment, the author wondered where the wallet connections end, location-wise. Although the analysis of the service distribution is valid for the considered moment, all services are terminated in the US. See Figure 10:3. Apple’s privacy statement [31] says that ‘Apple doesn’t store or have access to the credit, debit, or prepaid card numbers you added to Apple Pay’. However, this statement only refers to the sensitive payment data, not to other privacy-related information present during payment transactions at PoS or via remote payments.

6.6 Interpretation of Results

As expected, staying within legal limits, no weaknesses could be detected. However, we can state following:

ID	STRIDE If applicable	Description	Domain
1	S-----	The MiTM Attack applying SSL interception did not work, and rendered the wallet application, including the enrolment process, into failed status. There was no access granted to the enrolment data sent.	spoofing
2	S-----	Apple actively tests certificates for their validity via OSCP. At the same time, Apple verifies the signing root CA in its application for validity.	spoofing
3	---I--	Besides the loading of the card issuer picture, all connections used TLS to provide the necessary confidentiality for sent data.	confidentiality
4	-----D	The domain name services are mostly integrated with Akamai, which on either side allows the provision of the necessary content delivery performance, but also the necessary DDoS protection service.	denial of service
5	service location	All accessed services are in US. This implies that Apple must comply with European privacy and data protection laws. In 2015, the European Court of Justice invalidated the EC's Safe Harbour Agreement for data stored outside of the European Union. However, an evaluation of this is beyond the scope of the present work.	privacy data protection
6	enrolment	Enrolment follows the procedure described in the iOS Security Guide [34]	accuracy docu- mentation
7	enrolment	During enrolment, various destinations have been accessed that might deliver further meta data that is used later for fraud protection. These destinations are iTunes, device-assets, the iCloud, and location services.	meta data fraud protection
8	enrolment	Access to Mastercard's MDES services packet_id_37 and 44 are not documented.	accuracy docu- mentation

Table 6:4 Network Analysis–Overview Results

Chapter 7 Apple Pay and DSRP–How to Improve CNP

The following chapter describes how Apple Pay’s web payment implementation (DSRP) can be beneficial for reducing fraud in remote payment transactions (CNP). Current reports show that in countries where EMV chip technology has been introduced at PoS, criminals have shifted their efforts towards CNP fraud. In such areas, the fraud figures show a substantial rise in the CNP domain [61, 63]. The introduction of DSRP, with Apple Pay as an example, will play an important role in controlling and mitigating fraud figures. We do not consider in-app payment solutions or proprietary approaches that use dedicated merchant shopping applications.

7.1 Known Weaknesses of CNP Transactions

One main issue in the online payment process (CNP), where the merchant cannot physically identify that the legitimate cardholder is using their own credit card, is that we only have a limited set of options for verifying details that only the legitimate cardholder knows. That is, the ID&V process lacks the necessary strength. The same problem applies to the cardholder; there is no physical merchant identification available.

When it comes to remote payment transactions using a standard web browser application, the following major weaknesses are present, not only security-wise, but also in terms of the usability and manageability of the payment device:

ID	Description	Mitigation Method
1	The payment network mainly focuses on cardholder ID&V, but does not provide merchant identification to the customer.	none
2	The cardholder details we can verify are the <i>cardholder name</i> , the <i>card number</i> , the <i>card expiry date</i> , and the <i>card verification code</i> (CVV2) printed on the back of the credit card, and the <i>cardholder address</i> . For the latter, only the numerical values will be verified during AVS, because of the probability of spelling and keyboard errors.	fraud management
3	Fraud management is generally limited, as there is little meta data available regarding the payment device, location, device id, etc.	none
4	All cardholder details are potential targets of phishing attacks and social engineering.	none
5	The payment process is vulnerable to ‘man in the browser’ and ‘man in the middle’ attacks, which attempt to access payment credentials.	none
6	The credit card details are not protected via biometric or other strong authentication methods. A screenshot of the card is sufficient to use it for CNP fraud.	3D-Secure
7	CNP transactions do not use surrogate PANs in payment transactions. Compromised payment details finally lead to a time consuming and resource-intensive replacement process.	none
8	The payment process is interrupted when 3D-Secure authentication is processed. Thus, usability is decreased.	none

Table 7:1 CNP Transactions–CNP Weaknesses

7.2 Apple Pay–Digital Secure Remote Payment

Apple Pay is so far the only wallet supporting DSRP and using EMV chip technology to generate a payment cryptogram in its web payment process. Note that according to the Apple payment token reference [33], only Chinese markets use SE-generated EMV payment cryptograms [33], otherwise 3D-Secure cryptograms are used. One reason for employing the latter option could be that most merchants and payment processors already support the 3D-Secure packet structure. Apple Pay’s support of the 3D-Secure cryptogram may ease its adoption, but this is just speculation. We focus on the use of the EMV option, as this uses SE analogously to the contactless payment method.

Overview of payment model for Apple Pay web payments:

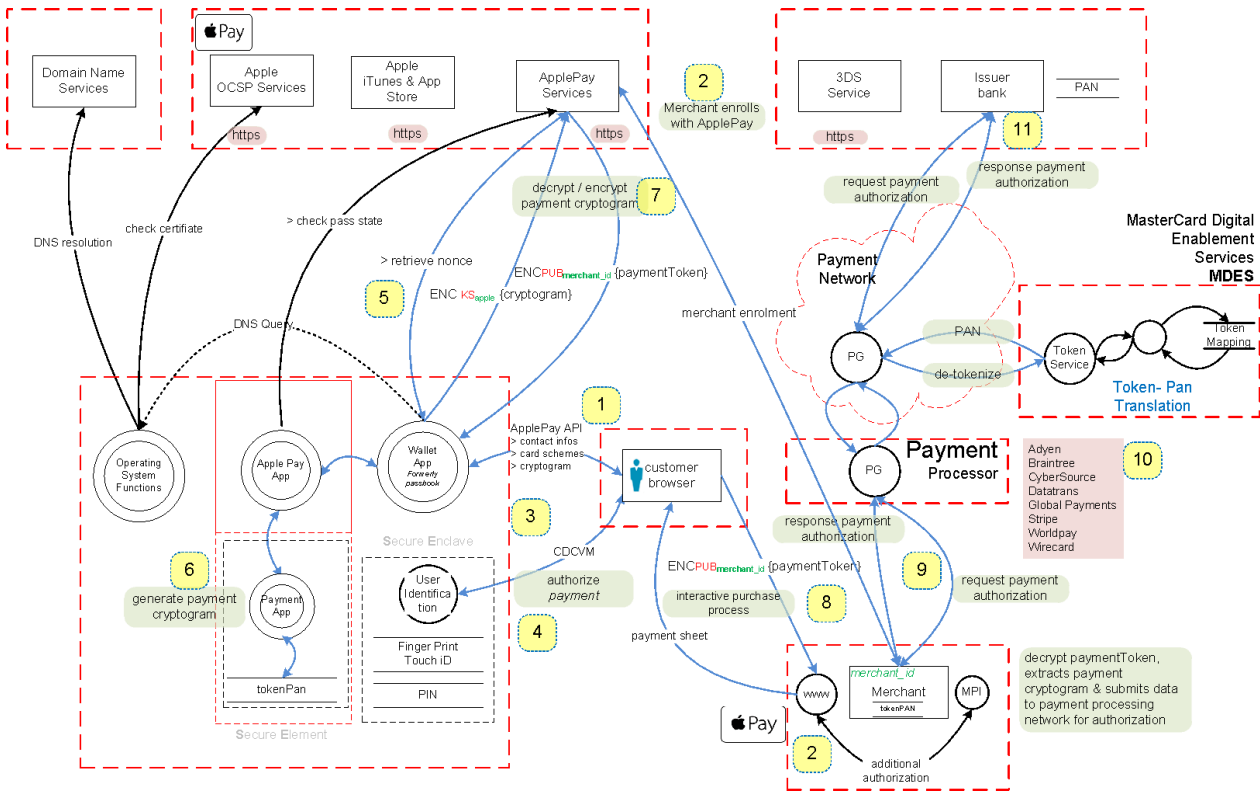


Figure 7:1 Apple Pay–Web Payment DSRP

Payment process: To offer Apple Pay as a payment method, a merchant must register with Apple Pay (2) and generate a public/private key pair. The enrolment process provides some degree of merchant identification and verification. Like the credit card enrolment into the wallet, this procedure needs to be assessed for its strength. The merchant application checks (3) whether Apple Pay is available on the customer’s smartphone (3). If available, the checkout process delivers the ‘payment sheet’ (see Figure 10:9) to the customer (1,3), who authorises the transaction via user identification (4). After payment authorisation, the wallet application receives a cryptographic nonce (5) from Apple Pay. This nonce will serve as a reply-protection mechanism. Next, the SE (6) generates the EMV payment cryptogram, which includes some other transaction data, such as merchant_id and the cryptographic nonce. This data (payment token) is finally encrypted, and sent to Apple Pay (see the figure below). This payment information is *confidentiality protected*, and can only be decrypted by Apple Pay.

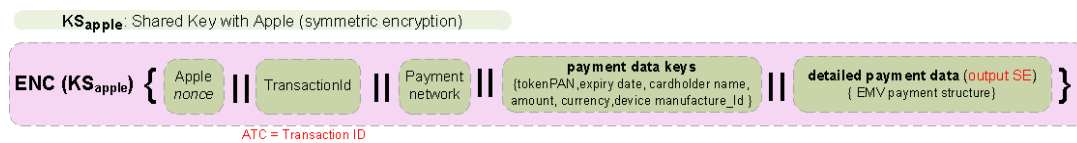


Figure 7:2 Apple Pay Web Payment–PaymentToken to Apple Pay

Information derived from Apple’ IOS Security Guide [34] and the corresponding Payment Token Format Reference [33] contradict on how the payment token is compiled. I adhered to the latter source used by the developers to show the token’s composition.

Apple Pay decrypts the data, encrypts it with the public key of the merchant (see figure below) and signs it. Then, this is sent back to the smartphone (7), which in turn forwards (8) the encrypted payment token to the merchant. The signature contains a timestamp, which helps to prevent reply attacks. If a specified time window has expired, then the token will not be accepted. The graphic shows, how the signature envelops the encrypted payment data and add some additional information. We provide confidentiality protection for the payment data and data origin authentication by adding the signature.

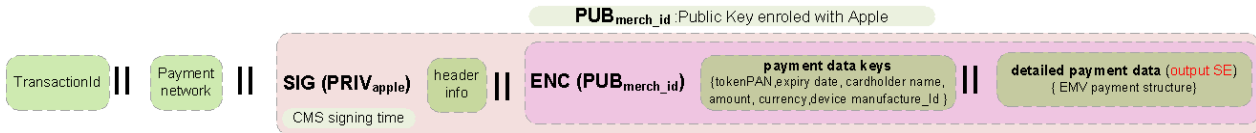


Figure 7:3 Apple Pay – Web Payment–PaymentToken to Merchant

The merchant decrypts the payment token, and forwards it via the associated payment processor for issuer authorisation (8, 9, 10, 11). To ease integration for the merchants, the payment processor often offers the decryption service.

7.3 How Apple Pay’s Approach Could Improve the Security of CNP Transactions

Apple Pay’s web payment (DSRP) introduces several security mechanisms that are missing in standard CNP transactions, and provides a significant security improvement.

Apple Pay’s main benefits for CNP transactions:

- **Merchant Identification:** Apple serves as trusted third party, and provides merchant identification through its merchant enrolment policy and the decryption and re-encryption process described in Chapter 7.2.
- **EMV Strength:** The SE generates an EMV-strength cryptogram. A captured payment token is useless, owing to domain restrictions, tokenisation, and the dynamic key generation used to construct the MAC of the cryptogram. This prevents cross channel fraud. MiTM and MiTB attacks are largely rendered useless.
- **Replay Prevention:** The use of nonces, CMS signing time (attribute (SigningTime ::= Time) referenced in RFC 5652 [79]), and merchant identification play an important role in preventing replay attacks.
- **Fraud Management–Meta Data:** The payment process includes far more meta data for supporting fraud analytics than a standard CNP transaction. This includes meta data related to Apple purchases, device information, tokenPAN information, and transaction information. See Figure 2:8 .
- **Payment Device Life Cycle Management:** In case a digitised credit card needs to be replaced or disabled, issuers can instantly cancel the card and prompt the user for a new enrolment, without the cost of a physical replacement.
- **Strong Cardholder Authentication:** The Consumer Device Cardholder Verification Method (CDCVM) supports two-factor device authentications, where one factor can be fingerprint authentication. This is comparable with the 3D-Secure cardholder verification and payment authorisation.
- **Seamless Experience:** Apple Pay provides a seamless payment experience in comparison with the 3D-Secure verification, where the consumer gets redirected. Note that 3-D Secure Version 2 tackles some of these usability issues by introducing the ‘EMV 3-D Secure Mobile SDK’ [80], which introduces a client-side software component to connect to the 3-D Secure ecosystem.
- **Encryption:** Sensitive payment data is confidentiality protected not only via transport channel (e.g. TLS) but also by explicit data field encryption (8). Hence, even a successful MiTM attack cannot access unprotected data.

Note: Apple Pay’s merchant identification constitutes feature that has been missing in the past. This will reduce, but not prevent, the setup of fraudulent merchant sites. The security of the merchant enrolment will be determined by the strength of the ID&V process. If the process is weak, we may see the occurrence of similar issues to Apple Pay’s ‘yellow path enrolment’ [8].

Chapter 8 Conclusion and Future Work

The analysis of the three eWallet solutions with respect to contactless payments at PoS has shown that in terms of the endpoints, i.e., the smartphone as a payment device, they differ in the way that they apply tokenisation. Apple Pay uses its SE to store the digitised PAN, cryptographic keys, and their issuer payment programs. This approach adheres to the principle of contactless credit cards using EMV chip technology. Samsung Pay and Android Pay use HCE technology, where their SE elements reside in the cloud, and replenish the eWallets with a set of dynamic keys. The solutions that use HCE may have usability issues in cases where they are not able to connect to the cloud when token replenishment is required. However, the extent to which this will slow down their adoption and whether this is relevant cannot be estimated.

The contactless mobile payment process will surely introduce new threats, and the strength of the initial card enrolment using the ID&V processes and the methods of authentication during transaction processing will be a key differentiator and enabler. However, in comparison with contactless payment methods (e.g., PayPass and payWave), the wallet solutions introduce stronger cardholder authentication and verification, lower handling costs for issuers, and additional meta data for effective fraud management, to name just a few benefits.

However, deeper investigation indicates that the introduction of tokenisation for the contactless payment process and the new EMV-supporting infrastructure in the form of digital enablement services points towards possible future changes in CNP transaction technology. Apple Pay's method of introducing DSRP from pure in-app payments into the world of the connected devices, such as iPads or other devices running Mac OSX, in order to facilitate EMV-strength payments, seems very promising to the author. This solution also offers a separate payment path from the device on which the merchant's application runs, which renders MiTM and MiTB attacks less effective.

In conclusion, we have now the possibility to move forward security-wise, while extending EMV technology to CNP payments using smartphones with a mobile browser, or to non-smartphone devices (e.g., iPads, laptops, desktops) using standard browsers to access the merchant's online tools and conduct secure EMV-strength payments. The following two proposed future projects aim towards this goal.

First future project: Apple Pay's trust model for identifying and securing payment information sent to the merchant should be transported globally to a CNP web payment process when using the mobile phone as payment device. This would be very similar to the failed SET (secure electronic transactions) approach [2, 3] that was employed many years ago, with the difference that this time the user does not take on the burden of key management. However, the cryptographic key management will still play a major role. The author would be particularly interested in seeing how identity-based encryption IDE, as proposed by Boneh and Franklin [88], could be introduced into the payment network infrastructure to provide end-to-end encryption between the cardholder and merchant via the use of a public key generator (PKG). The thought behind 'identity-based encryption' is to allow any party to generate a public key using a known identity value, such as an ASCII string. The known identity value could be, for example, the merchant web domain name combined with a time stamp and the amount purchased (e.g., *.merchant.com || time-stamp || amount), which could produce unique dynamic keys. IDE involves a trusted third party (private key generator), who would possess the key escrow capability (key recovery). This is often an undesired security property, because of non-repudiation issues. However, it would serve the purpose of solving transaction disputes.

Merchant domain ranges would point to the scheme allocated service providers (PKG), which the merchants access to retrieve their private keys. This delegation construct would allow the spreading of the global load to different PKGs, similarly to DNS domain delegation.

Advantages:

- Use of EMV chip technology for remote payments via the internet, tackling ongoing CNP fraud.
- Immediate encryption for data sent to merchants, without previous key distribution.
- End-to-end encryption of cryptogram between customer and merchant. This could also be applied to contactless CP payments to achieve end-to-end encryption on the NFC channel.
- Identification of merchants to the PKG (trusted third party) and customer.
- Solution is not Apple-centric—this would be available for others.

Disadvantages - Challenges:

- PKG service must be online and available.
- Key derivation functionalities needed for the merchant, in order to provide the necessary speed of transactions.
- Key escrow functionality of PKG.
- Key lifecycle management for different use cases, expiry, replay protection, etc.
- Merchant enrolment.
- API must be developed.
- Introduction of PKG as a new stakeholder into the payment infrastructure.
- Distribution of PKG services within a payment network.
- Integrating HCE for devices not providing an SE.

The simplified flow, using the Apple Pay infrastructure as before, could look as follows:

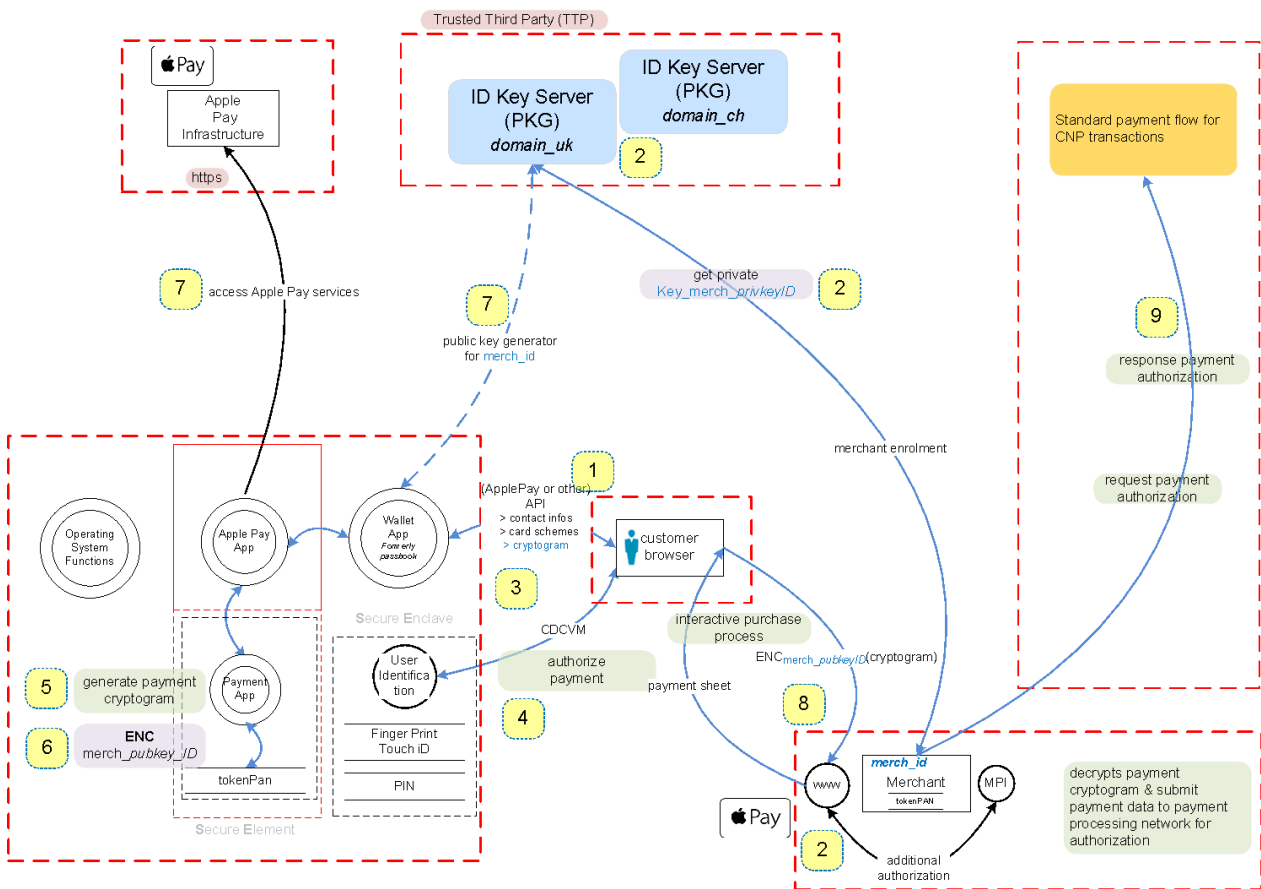


Figure 8:1 Future Project – Introduction of IDE

How it should work: Customer/cardholder receives the payment sheet (1,3) from the merchant, who has previously enrolled with the PKG (private key server). The cardholder authorises the payment sheet (4), and the wallet application generates the EMV cryptogram (5) and encrypts it with the *merch_pubkeyID* (6), derived via the IDE technique (7). The encrypted payment cryptogram is sent to the merchant (8) via a browser API, and from there is sent further up for normal payment processing (9).

Second future project: Another area for future work involves facilitating mobile payments via standard web browsers on a third-party device, but using the mobile phone as the payment device where the EMV cryptogram is generated. The solution should be universal, and not bound to a single vendor technology. This proposal is an extension to the first future project, and re-uses its components.

The challenge: Because the payment device is not the same as the device being used for the shopping, we need to provide a means to couple the two entities and involve a trusted third party to exchange the encrypted cryptograms.

The author's thoughts: First, the merchant would provide his payment sheet to the customer via a QR code or similar graphical representation. The customer and cardholder would scan the QR code with their payment device that hosts the wallet application, and verify the payment sheet. On approval, the payment cryptogram would be generated and encrypted with the public key of the merchant, using the IDE as in the previous solution. Next, the trusted third party, which could also be the PKG, would forward the encrypted cryptogram to the merchant. The merchant decrypts the packet, and using the payment information forwards it to its upstream payment processor for normal payment processing via the standard procedure. See Chapter 2.3.

Additional advantages:

- We provide EMV-strength payment cryptograms with remote payments, where the device for used for shopping is not the payment device.
- We use a second channel to provide and transport the payment cryptograms, achieving the separation of payment and shopping.

Disadvantages/challenges:

- Availability and setup of a trusted third party, who forwards the encrypted payment cryptograms.
- The security aspects of the introduction of a QR codes or alternative transport mechanism.
- Transaction speed and user experience during the payment process.

The simplified flow could look as follows:

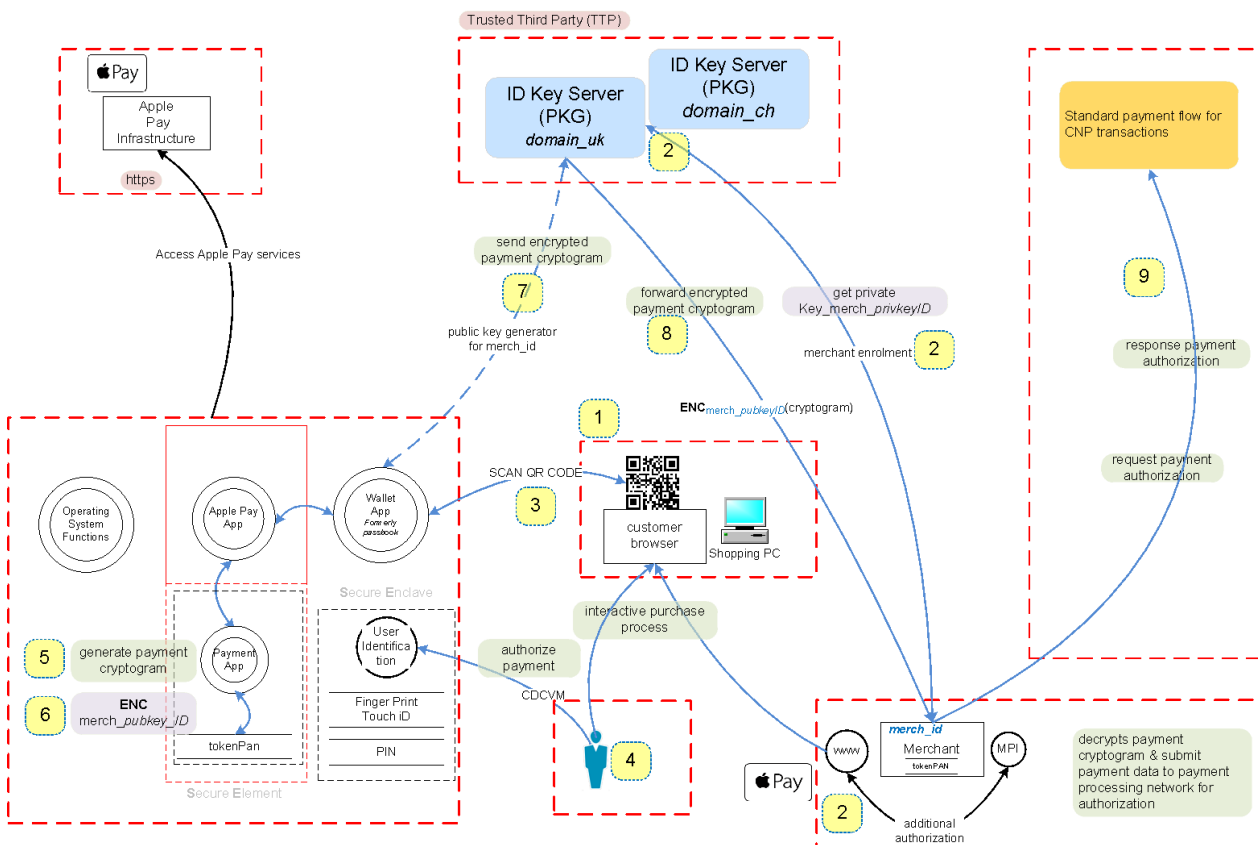


Figure 8:2 Future Project—Introduction of Device Independency

How it works: Instead of having the wallet and the browser on the same device, the merchant's web application (2) sends a QR code containing the payment sheet to the screen of the customer (1). The cardholder scans the QR code (3) using their payment device (3), and verifies and authorises the payment sheet (4). Then, the wallet application generates the EMV cryptogram (5) and

encrypts it with the `merch_pubkeyID` **(6)** derived via the IDE technique **(7)**. The encrypted payment cryptogram is now sent to the PKG or another TTP, who forwards the packet to the merchant **(8)** via a dedicated API, and from there further up for normal payment processing **(9)**.

Chapter 9 Bibliography

- [1] THREE BILLION TOUCHES IN THE LAST 12 MONTHS, May 13, 2016-last update, Europeans "touched to pay" three billion times in the last 12 months, says Visa Europe [Homepage of Shillito Market Intelligence Ltd], [Online]. Available: https://www.visaeurope.com/newsroom/news/european_used_contactless_3_billion_times_last_year.
- [2] JARUPUNPHOL, P. and BUATHONG, W., 2013. Secure Electronic Transactions (SET): A Case of Secure System Project Failures. *International Journal of Engineering and Technology*, 5(2), pp. 278.
- [3] ISMAILI, H.E., HOUMANI, H. and MADROUMI, H., 2014. A Secure Electronic Transaction Payment Protocol Design and Implementation. *International Journal of Advanced Computer Science and Applications*, 5(5), pp. 172-180.
- [4] ZHONG, J., DHIR, A., NIEMINEN, M., HÄMÄLÄINEN, M. and LAINE, J., 2013. Exploring Consumer Adoption of Mobile Payments in China, 2013, ACM, pp. 318-325.
- [5] DAHLBERG, 2015. Mobile Payments in the Light of Money Theories: Means to Accelerate Mobile Payment Service Acceptance? 2015, ACM, pp. 1-8.
- [6] ECB, January 31, 2013-last update, European Central Bank - Recommendations for the Security of Internet Payments: Final version after public consultation [Homepage of European Central Bank], [Online]. Available: http://gateway.proquest.com/openurl?url_ver=Z39.88-2004&res_dat=xri:policyfile&rft_dat=xri:policyfile:article:00154030.
- [7] POURGHOMI, P., ABI-CHAR, P.E. and GHINEA, G., 2015. Towards a Mobile Payment Market: A Comparative Analysis of Host Card Emulation and Secure Element. *International Journal of Computer Science and Information Security*, 13(12), pp. 156.
- [8] ARSTECHNICA, March, 2015-last update, Fraud on Apple Pay is identity-theft based. Available: <http://arstechnica.com/apple/2015/03/the-weak-link-in-apple-pays-strong-chain-is-bank-verification-whos-to-blame/>.
- [9] PCI SECURITY STANDARDS COUNCIL, 2011-last update, PCI Data Security Standard (PCI DSS) - PCI DSS Tokenization Guidelines. Available: <https://www.pcisecuritystandards.org/>.
- [10] PCI SECURITY STANDARDS COUNCIL, April, 2015-last update, PCI Data Security Standard (PCI DSS) - Guideline: Tokenization Product Security Guidelines Version: 1.0;. Available: www.pcisecuritystandards.org.
- [11] VERIZON, 2016. Verizon Data Brach Investigations Report 2016. Verizon DBIR 2016.
- [12] VISA, , VISA Token Service VTS. Available: <https://developer.visa.com/products/vts> [March 13, 2017].
- [13] PCI SECURITY STANDARDS COUNCIL, Dec, 2015-last update, Additional Security Requirements and Assessment Procedures for Token Service Providers (EMV Payment Tokens) Version 1.0. Available: https://www.pcisecuritystandards.org/documents/FAQs_for_TSP_Requirements_v1.pdf.
- [14] BSI ISO STANDARD, 2015. BS ISO/IEC 7812-1:2015: Identification cards. Identification of issuers. Numbering system. British Standards Institute.
- [15] CREDIT CARD VALIDATOR, , Credit Card Validator. Available: <https://www.creditcardvalidator.org> [March 13, 2017].
- [16] CAUSLEY, B., 2012. The secret behind the Luhn-IE. XRDS: Crossroads, The ACM Magazine for Students, 19(1), pp. 81-82.
- [17] CLULEY, G., 2015. How Apple Pay can make Credit Card Fraud Easier. ACI Information Group.
- [18] EMVCO, 2014-last update, EMV® Payment Tokenisation Specification - Framework Version 1.0. Available: <https://www.emvco.com/>
- [19] MASTERCARD, , Mastercard Digital Enablement Service MDES . Available: <https://developer.mastercard.com/product/mdes> [Mar 13, 2017].
- [20] APPLE SUPPORT and HT202527, Dec 2, 2016-last update, Apple Pay - How do I accept Apple Pay in my store - HT202527. Available: <https://support.apple.com/en-us/HT202527>.
- [21] EMVCO and BOOK 2 KEY MANAGEMENT, Nov, 2011-last update, EMVCo Integrated Circuit Card Specifications for Payment Systems - Book 2 Security and Key Management, Version 4.3. Available: www.emvco.com.
- [22] ANDROID PAY - SUPPORT DSRP, Dec 15, 2015-last update, Android Pay Adds In-app Purchasing Feature, Catches up to Apple Pay. Available: <http://arstechnica.com/gadgets/2015/12/android-pay-adds-in-app-purchasing-feature-catches-up-to-apple-pay/>.
- [23] MC DIGITAL SECURE PAYMENTS, Sep 10, 2014-last update, DSRP - Digital Secure Payments. Available: <http://newsroom.mastercard.com/2014/09/10/mastercard-digital-enablement-service-mdes-making-digital-payments-happen/>.
- [24] SAMSUNG, K., Samsung KNOX Security Framework. Available: <http://developer.samsung.com/tech-insights/pay/device-side-security> [Mar 13, 2017].
- [25] SAMSUNG DEVELOPER TOKENIZATION, Samsung Pay Tokenization and Transaction Cryptograms. Available: <http://developer.samsung.com/tech-insights/pay/tokenization> [Mar 13, 2017].

- [26] APPLE PAY - GERMAN DEVELOPERS and CARSTEN EILERS, Aug 27, 2015-last update, Apple Pay - German Developers - How it Works. Available: <https://entwickler.de/online/security/apple-pay-sicherheit-funktion-169792.html>.
- [27] APPLE PAY - PARTICIPATING BANKS IN EUROPE, Mar 9, 2017-last update, Apple Pay Participating Banks in Europe. Available: <https://support.apple.com/en-gb/HT206637>.
- [28] APPLE PAY - CONSUMER DEVICE CARDHOLDER VERIFICATION METHOD, Dec 2, 2016-last update, Apple Pay - Consumer Device Cardholder Verification Method . Available: <https://support.apple.com/en-us/HT202527>.
- [29] APPLE SUPPORT, Jul 29, 2014-last update, Technical Note TN2232 HTTPS Server Trust Evaluation. Available: https://developer.apple.com/library/prerelease/content/technotes/tn2232/_index.html.
- [30] APPLE - PKI REFERENCE, Apple - PKI Reference. Available: <https://www.apple.com/certificateauthority/> [Mar 13, 2017].
- [31] APPLE - SECURITY AND PRIVACY OVERVIEW, Dec 2, 2016-last update, Apple Pay - Security and Privacy Overview. Available: <https://support.apple.com/en-us/HT203027>.
- [32] APPLE - ABOUT IN-APP PURCHASE, Oct 21, 2015-last update, Apple Pay - About In-App Purchase. Available: <https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/StoreKitGuide/Introduction.html>.
- [33] APPLE - PAYMENT TOKEN FORMAT REFERENCE, Dec 1, 2016-last update, Apple Pay - Payment Token Format Reference. Available: https://developer.apple.com/library/content/documentation/PassKit/Reference/PaymentTokenJSON/PaymentTokenJSON.html#//apple_ref/doc/uid/TP40014929.
- [34] APPLE IOS SECURITY, May, 2016-last update, Apple - IOS Security - iOS 9.3 or later. Available: https://www.apple.com/business/docs/iOS_Security_Guide.pdf.
- [35] APPLE PAY COMPATIBILITY, Jan 3, 2017-last update, Apple Pay is compatible with these devices . Available: <https://support.apple.com/nl-be/KM207105>.
- [36] JAVA CARD PLATFORM, Java Card Platform. Available: <https://javacardforum.com/resources/faqs/> [Mar 13, 2017].
- [37] ANDROID - FINGERPRINT HAL - TEE, Android - Fingerprint HAL - TEE. Available: <https://source.android.com/security/authentication/fingerprint-hal.html> [Mar 13, 2017].
- [38] GOOGLE, Participating Banks and Supported Cards for Android Pay . Available: <https://support.google.com/androidpay/answer/6314169> [Mar 13, 2017].
- [39] ANDROID, Android Pay - ID&V Option during Card Enrolment. Available: <https://support.google.com/androidpay/answer/6289372-verify> [Mar 13, 2017].
- [40] ANDROID PAY - HOST-BASED CARD EMULATION (HCE), Android Pay - Host-based Card Emulation (HCE). Available: <https://developer.android.com/guide/topics/connectivity/nfc/hce.html> [Mar 13, 2017].
- [41] BALFE, S. and PATERSON, K., 2008. e-EMV: Emulating EMV for Internet Payments with Trusted Computing Technologies, 2008, ACM, pp. 81-92.
- [42] UL - MOBILE CLOUD-BASED PAYMENT SECURITY EVALUATION, Mobile Cloud-based Payment Security Evaluation. Available: <http://services.ul.com/service/mobile-cloud-based-payment-security-evaluation> [Mar 13, 2017].
- [43] JULIEN, M., Dec 30, 2016-last update, EMV Reader Application for Android Version 4.2.4 . Available: <https://play.google.com/store/apps/details?id=com.github.devniied.emvnfccard&hl=en> [Mar 13, 2017].
- [44] ANDROID DEVELOPERS, Keep your Payment Info Safe. Available: https://support.google.com/androidpay/answer/6289406?visit_id=1-636196560296708108-560677702&rd=1 [Mar 13, 2017].
- [45] GOOGLE, Android Pay - Supported Devices - Set up Android Pay. Available: <https://support.google.com/androidpay/answer/6224811> [Mar 13, 2017].
- [46] MICHIELS, W., GORISSEN, P., 2007. Mechanism for Software Tamper Resistance: an Application of White-box Cryptography, 2007, ACM, pp. 82-89.
- [47] SAMSUNG DEVELOPER, Samsung Pay - Remote Management. Available: <http://developer.samsung.com/tech-insights/pay/remote-management> [Mar 13, 2017].
- [48] SAMSUNG-CDCVM, Samsung - User Identity Setup and Credential Verification. Available: <http://developer.samsung.com/tech-insights/pay/user-identity-setup-and-credential-verification> [Mar 13, 2017].
- [49] SAMSUNG-TEE, Samsung - Device-side Security Samsung Pay, TrustZone, and the TEE. Available: <http://developer.samsung.com/tech-insights/pay/device-side-security> [Mar 13, 2017].
- [50] SAMSUNG-TOKEN HANDLING, Samsung - Token Handling by Samsung Pay. Available: <http://developer.samsung.com/tech-insights/pay/token-handling-by-samsung-pay> [Mar 13, 2017].
- [51] KNOX SECURITY, Samsung KNOX - Certifications. Available: <https://www.samsungknox.com/en/knox-technology/security-certifications> [Mar 13, 2017].
- [52] SAMSUNG PAY - DSRP SUPPORT, Oct 25, 2016-last update, Samsung Pay Adds Online and In-app Purchases. Available: <http://www.androidcentral.com/samsung-pay-expands-new-countries-adds-online-and-app-purchases>.

- [53] SAMSUNG PAY ENROLMENT, Samsung Pay - Enrolment. Available: <http://www.samsung.com/us/support/answer/ANS00045081/> [Mar 13, 2017].
- [54] SAMSUNG PAY - SUPPORTED DEVICES AND SOFTWARE, Samsung Pay - Device Compatibility. Available: <http://www.samsung.com/us/support/answer/ANS00045945/> [Mar 13, 2017].
- [55] SAMSUNG PAY - SUPPORTED ISSUERS, Samsung Pay - Supported Issuers. Available: <http://www.samsung.com/us/samsung-pay/compatible-cards/> [Mar 13, 2017].
- [56] FINANCIAL FRAUD ACTION UK, 2017-last update, FRAUD THE FACTS 2016 - CNP Figures. Available: https://www.financialfraudaction.org.uk/fraudfacts16/assets/fraud_the_facts.pdf [Mar 13, 2017].
- [57] CHECKPOINT, Aug 16, 2015-last update, Mobile Threat Prevention to Secure the Mobile Endpoints. Available: <https://www.checkpoint.com/press/2015/check-point-launches-mobile-threat-prevention-to-secure-the-mobile-enterprise/>.
- [58] KREBS ON SECURITY, Aug 8, 2016-last update, Data Breach at Oracle's MICROS Point-of-Sale Division. Available: <http://krebsonsecurity.com/2016/08/data-breach-at-oracles-micros-point-of-sale-division/>.
- [59] CHRISTIAN KILLER, CHRISTOS TSIARAS, BURKHARD STILLER, June, 2015-last update, An Off-the-shelf Relay Attack in a Contactless Payment Solution. Available: https://files.ifi.uzh.ch/CSG/staff/tsiaras/Extern/Theses/VA_ChristianKiller.pdf.
- [60] ENISA-MOBILE PAYMENT SECURITY, Dec, 2016-last update, Security of Mobile Payments and Digital Wallets. Available: <https://www.enisa.europa.eu/publications/mobile-payments-security/>.
- [61] CROWE M., PANDY S., FEDERAL RESERVE BANK OF BOSTON, Nov 10, 2016-last update, Assessing Card-Not-Present Fraud in the Mobile Payments Environment. Available: <https://www.bostonfed.org/publications/mobile-payments-industry-workgroup/getting-ahead-of-the-curve-assessing-card-not-present-fraud-in-the-mobile-payments-environment.aspx>.
- [62] ECB DRAFT, Nov, 2013-last update, RECOMMENDATIONS FOR THE SECURITY OF MOBILE PAYMENTS. Available: <https://www.ecb.europa.eu/paym/cons/pdf/131120/recommendationsforthesecurityofmobilepaymentsdraftpc201311en.pdf>.
- [63] PANDY S., Feb 12, 2016-last update, Mitigating Fraud Risk in the Card-Not-Present Environment. Available: <https://www.bostonfed.org/publications/mobile-payments-industry-workgroup/mitigating-fraud-risk-in-the-card-not-present-environment.aspx>.
- [64] OWASP, Sep 28, 2015-last update, OWASP Top Ten Cheat Sheet. Available: https://www.owasp.org/index.php/OWASP_Top_Ten_Cheat_Sheet.
- [65] METHODOLOGIES, PRACTICES and TOOLS TO ENABLE A FUNCTIONALLY INTEGRATED CYBER SECURITY ORGANIZATION, A Threat-Driven Approach to Cyber Security. Lockheed Martin.
- [66] POTTER, B., 2009. Microsoft SDL Threat Modelling Tool. Network Security, 2009(1), pp. 15-18.
- [67] NIST, Jan, 2017-last update, NIST Cyber Security Framework CSF Version 1.1. Available: <https://www.nist.gov/cyberframework>.
- [68] OWASP, Jan 19, 2017-last update, Pinning Cheat Sheet. Available: https://www.owasp.org/index.php/Pinning_Cheat_Sheet.
- [69] TREND MICRO, Sept 21, 2015-last update, Malware-Laced Xcode Tool Used to Infect iOS Apps. Available: <https://www.trendmicro.com/vinfo/us/security/news/mobile-safety/malware-laced-xcode-tool-used-to-infect-ios-apps>.
- [70] MCAFEE, McAfee Webgateway. Available: <http://www.mcafee.com/us/products/web-gateway.aspx> [Mar 13, 2017].
- [71] APPLE PAY, Apple Pay in Switzerland - Supported Issuers. Available: <http://www.apple.com/chde/apple-pay/> [Mar 13, 2017].
- [72] VMWARE, VMware Fusion for MAC
. Available: <http://www.vmware.com/products/fusion.html> [Mar 13, 2017].
- [73] GNU WIRESHARK, Wireshark packe. Available: <https://www.wireshark.org> [Oct 18, 2016].
- [74] EFT LAB, EFT EMV Transaction Services Tools. Available: <https://eftlab.co.uk/index.php/downloads/bp-tools#> [Oct 18, 2016].
- [75] ACS, ACR123U Intelligent Contactless Reader Software Development Kit. Available: <http://www.acs.com.hk/en/products/346/acr123u-intelligent-contactless-reader-software-development-kit/> [Nov 10, 2016].
- [76] JULIEN, M. and VERSION, 4.2.4., 2016. Credit Card Reader NFC (EMV) for Android. Google Play Store: .
- [77] OFFENSIVE SECURITY, Kali Linux Advanced Penetration Testing Distribution
. Available: <https://www.kali.org/downloads/> [Oct 14, 2016].
- [78] LEGISLATION, Computer Misuse Act 1990. Available: <http://legislation.data.gov.uk/ukpga/1990/18/data.htm?wrap=true> [Nov 18, 2017].
- [79] RFC 5652, RFC 5652 Cryptographic Message Syntax (CMS)
. Available: <https://tools.ietf.org/html/rfc5652#page-41> [Mar 13, 2017].
- [80] EMVCO, Jan, 2017-last update, EMVCo - 3-D Secure – SDK Specification Version 2.0.0. Available: <https://www.emvco.com/specifications.aspx?id=299>.
- [81] EMVCO, Jan, 2017-last update, EMVCo - 3-D Secure - SDK - Device Information Version 2.0.0
. Available: <https://www.emvco.com/specifications.aspx?id=299>.

- [82] PCI SECURITY STANDARDS COUNCIL, Apr, 2016-last update, PCI Data Security Standard (PCI DSS) - Requirements and Security Assessment Procedures Version 3.2. Available: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2.pdf?agreement=true&time=1486717090457.
- [83] EMVCO, Nov, 2015-last update, EMVCo Contactless Mobile Payment Handset Requirements for Contactless Mobile Payment, Version 1.1. Available: <http://www.emvco.com/>.
- [84] EMVCO, Jun, 2015-last update, EMVCo Contactless Mobile Payment - White Paper on Contactless Mobile Payment, Version 2.2. Available: <https://www.emvco.com>.
- [85] EMVCO, Mar, 2016-last update, EMVCo Contactless Specifications for Payment Systems -Architecture and General Requirements Book A, Version 2.6. Available: www.emvco.com.
- [86] EMVCO, Feb, 2016-last update, EMVCo Contactless Specifications for Payment Systems - Kernel 2 Specification Book C-2, Version 2.6. Available: <http://www.emvco.com/>.
- [87] EMVCO, Feb, 2016-last update, EMVCo Contactless Specifications for Payment Systems - Kernel 2 Specification Book C-3, Version 2.6. Available: <http://www.emvco.com/>.
- [88] BONEH, D. and FRANKLIN, M., 2003. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, 32(3), pp. 586-615.
- [89] DIAKOS, T.P., BRIFFA, J.A., BROWN, T.W.C. and WESEMAYER, S., 2013. Eavesdropping Near-field Contactless Payments: a Quantitative Analysis. *The Journal of Engineering*.
- [90] GOLLMANN, D., 2011. *Computer security*. 3. ed., 1. publ. edn. Hoboken, N.J: Wiley.
- [91] EUROPOL, , PAYMENT FRAUD - CNP Transactions. Available: <https://www.europol.europa.eu/crime-areas-and-trends/crime-areas/forgery-of-money-and-means-of-payment/payment-fraud> [Mar 12, 2017].
- [92] CROWE MARIANNE, Nov 10, 2016-last update, Getting Ahead of the Curve: Assessing Card-Not-Present Fraud. Available: https://www.frbatlanta.org/-/media/documents/rprf/rprf_pubs/2016/11-getting-ahead-of-the-curve-assessing-card-not-present-fraud-2016-11-18.pdf [Mar 13, 2017].
- [93] EUROPEAN BANKING AUTHORITY, Dec 19, 2014-last update, FINAL GUIDELINES ON THE SECURITY OF INTERNET PAYMENTS. Available: [https://www.eba.europa.eu/documents/10180/934179/EBA-GL-2014-12+\(Guidelines+on+the+security+of+internet+payments\)_Rev1](https://www.eba.europa.eu/documents/10180/934179/EBA-GL-2014-12+(Guidelines+on+the+security+of+internet+payments)_Rev1).
- [94] MASTERCARD, Sep 10, 2014-last update, MasterCard Digital Enablement Service (MDES): Making Digital Payments Happen. Available: <https://newsroom.mastercard.com> [Mar 13, 2017].
- [95] GLOBALPLATFORM FORUM, , GlobalPlatform made simple guide: Secure Element. Available: <https://www.globalplatform.org/mediaguideSE.asp> [Mar 13, 2017].
- [96] CROWE, M. and PANDY, S., May 10, 2016-last update, Understanding the Role of Host Card Emulation in Mobile Wallets. Available: <https://www.bostonfed.org/publications/payment-strategies/understanding-the-role-of-host-card-emulation-in-mobile-wallets.aspx> [Jan 16, 2017].

Additional Sources:

The references below have influenced the project but have not been cited within the text. They supported the research and helped to gain a stronger understanding of the topic.

- [a1] EPC - EUROPEAN PAYMENT COUNCIL, Apr 22, 2014-last update, EPC Comments on the Draft 'Recommendations for the Security of Mobile Payments'. Available: http://www.europeanpaymentscouncil.eu/index.cfm/newsletter/article/?articles_uid=56E7ACF8-5056-B741-DB4DCC0F6BD1D12E.
- [a2] ECB, Jan, 2013-last update, Recommendations for the Security of Internet Payments. Available: https://www.ecb.europa.eu/press/pr/date/2013/html/pr130131_1.en.html.
- [a3] EUROPEAN CENTRAL BANK, Jul, 2015-last update, Fourth Report on Card Fraud. Available: https://www.ecb.europa.eu/pub/pdf/other/4th_card_fraud_report.en.pdf.
- [a4] SMART CARD ALLIANCE PAYMENTS, Oct, 2012-last update, EMV and NFC: Complementary Technologies that Deliver Secure Payments and Value- Added Functionality.
- [a5] POURGHOMI, P. and GHINEA, G., 2013. Ecosystem scenarios for cloud-based NFC payments, 2013, ACM, pp. 113-118.
- [a6] BSI, Nov, 2016-last update, BAI - Hardening Guide for Samsung KNOX Version 1.0. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/Samsung_Knox/Samsung_Knox.pdf?__blob=publicationFile&v=7.
- [a7] SAMSUNG PAY, Samsung - Samsung Pay In-App Purchases . Available: <http://www.samsung.com/us/support/answer/ANS00061541/> [Mar 13, 2017].

[a8] SAMSUNG IN-APP PAYMENT, Samsung Pay Support for In-app Payments. Available: <http://www.samsung.com/us/support/answer/ANS00061541/> [Mar 13, 2017].

[a9] ANDROID DEVELOPERS, Nov 3, 2016-last update, Android Pay API. Available: <https://developers.google.com/android-pay/integration/payment-token-cryptography>.

[a10] APPLE - APPLE PAY PROGRAMMING GUIDE, Dec 12, 2016-last update, Apple Pay - Programming Guide. Available: https://developer.apple.com/library/content/ApplePay_Guide/.

Chapter 10 Appendix

Definitions and Abbreviations

What	Description
3D-Secure	3-Domain Secure is a secure communication protocol used to enable real-time cardholder authentication directly from the card issuer during an online transaction to improve online transaction.
Account takeover	Account takeover fraud occurs when a fraudster obtains an individual's bank or payment card number and other personal identifying information (PII), such as email, password, username, or social security number. The fraudster changes the contact information, or adds another user to an existing account, which they can then use to conduct transactions.
CDCVM	Consumer Device Cardholder Verification Method (CDCVM) is a type of consumer verification method (CVM) supported by card networks when assessing transactions originating from mobile devices. Verification is used to evaluate whether the person presenting the payment instrument is the legitimate owner of the instrument, and affects where the liability lies for fraudulent transactions. The definition has been derived from [20].
CNP	Card-not-present payment occurs when a cardholder/card is not physically present when making a purchase, preventing the merchant from validating the cardholder as the card owner. Examples of CNP payments include internet payments, telephone, or mail order. The definition has been derived from [63].
CNP Fraud	CNP fraud involves the unauthorised use of payment credentials (stolen credit/debit card number) to purchase products or services in a non-face-to-face environment between the customer and the merchant, such as an e-commerce transaction via a call centre, computer, mobile device, or mail order. The definition has been derived from [91].
CoF	Card-on-file (CoF) is the authorised storage of a consumer's payment credentials by a merchant or payment service provider that allows the consumer to make repeat or automatic payments, including money transfers, without the need to re-enter payment credentials each time. The definition has been derived from [92].
Credentials	The personal and confidential information provided for the purposes of authentication. Credentials can also refer to the physical tool used for obtaining the information (e.g., one-time-password generator or smart card), or to something the user memorises or represents (such as biometric characteristics). The definition has been derived from [93].
DSRP	A DSRP transaction is a payment method specification that uses EMV-like cryptography, achieving the same security level for mobile commerce as in a "card present" environment. Digital Secure Remote Payment is a transaction method where a consumer can make in-app purchases using a token. While contactless payments leverage NFC technology for point-of-sale (POS) transactions, DSRP delivers EMV-like transactions for in-app payments. Tokens are card numbers that mobile devices use in the place of the card number printed on the plastic. MDES validates the transaction, maps from the token back to the PAN, and forwards it to the issuer for authorisation. The definition has been derived from [94].
EMV ID&V	A valid method, through which an entity may successfully validate the cardholder and the cardholder's account to establish a confidence level for payment token to PAN/cardholder binding. This is crucial during the card enrolment process, before a tokenPAN is added to the eWallet.
eWallet digital wallet	A software-based container that allows a user to store personal information (e.g., ID, insurance, health, transportation, etc.), loyalty and couponing information, and payment information (i.e., credit card or bank account) that can be used to perform e-commerce/m-commerce transactions. The wallet application may reside on the user's mobile device or computer.
Global Platform	Defines a secure element (SE) as a tamper-resistant one-chip secure microcontroller capable of securely hosting applications and their cryptographic data (e.g., key management) in accordance with the rules and security requirements set forth by trusted authorities. The definition has been derived from [95].
Host Card Emulation (HCE)	A software-based technology that supports the ability for a mobile wallet app to run on the host processing unit of a mobile device, in order to communicate through the NFC controller in the mobile device to a contactless NFC-enabled POS terminal/reader to pass payment card credentials (or payment token), eliminating the need to access payment credentials or tokens stored on the physical SE chip in a mobile device. The definition has been derived from [96].
In-APP	In-app payments are remote CNP payments conducted within an online application running on a smart phone. In comparison to DSRP, this may use card on file PAN data and not apply EMV-strength transaction processing using cryptographic means.
in-app ID&V method	The user may be instructed to find a code in their mobile banking app and verify the amount. The card network, acting as the merchant, sends the authorisation to the issuer, and then reverses the transaction within a specific timeframe. This allows the authentication of the cardholder on a second channel.
MITB	Man in the browser(MITB) is a type of MITM attack where the attacker exploits vulnerabilities in the browser software to implant malware. The implanted Trojan can be used for various purposes, such as to inject or listen to payment transactions or enforce redirection to malicious websites.
MITM	A man-in-the-middle (MITM) attack intercepts a communication between two entities. For example, an attacker within reception range of an unencrypted Wi-Fi wireless access point can insert himself in the communication between the two points.
MNO	The mobile network operator is responsible for providing the GSM network for data transmission. It also deals with the life cycle management of NFC ecosystems, as well as data provisioning over-the-air (OTA). MNO is the SE issuer, because SE takes the form factor of UICC.
PAR	The payment account reference is a unique identifier associated with a specific cardholder PAN. This can be used as a reference for all cardholder transactions to leverage for other value added services, such as loyalty. The definition has

What	Description
	been derived from EMVCo [83].
PCI SSC	PCI Security Standards are technical and operational requirements set by the PCI Security Standards Council (PCI SSC) to protect cardholder data. The standards apply to all entities that store, process, or transmit cardholder data, with requirements for software developers and manufacturers of applications and devices used in those transactions. The definition has been derived from PCI Council [9].
PCI DSS	PCI DSS regulates merchants and service providers in term of a secure environments. The definition has been derived from PCI Council [9].
PoS payment	Payment where the payer or originator is physically present at the merchant's physical location.
PSP	A payment service provider (PSP) may be a payment processor, merchant acquirer, gateway, wallet provider, or other type of third party that serves as an intermediary between the merchant and the payment network.
Secure Element SE	A certified tamper-resistant platform (device or component), capable of securely hosting applications and their confidential and cryptographic data (e.g., key management) in accordance with the rules and security requirements set forth by a set of well-identified trusted authorities. Examples include the UICC, an embedded secure element, a chip card, and an SD card. The definition has been derived from [92].
Sensitive payment data	We use the definition of the European Central Bank [62], "Data that could be used to carry out fraud, excluding the name of the account owner and the account number, including data enabling a payment order to be initiated (e.g., PAN, card expiry date, CVx2), data used for authentication (customer identifiers, birth date, passwords, codes, PIN, secret questions, passwords/codes for reset, telephone number, certificates), data used for ordering payment instruments, or authentication tools to be sent to customers (customer's physical address, telephone number, e-mail address), as well as data, parameters, and software that, if modified, may affect the legitimate party's ability to verify payment transactions, authorise e-mandates, or control the account (such as 'black' and 'white' lists and customer-defined limits), and browser plug-ins and java applets provided by PSPs to their customers."
tcpdump	Tcpdump prints out a description of the contents of packets on a network interface.
TEE	The trusted execution environment (TEE) is a secure area of the main processor of a smart phone (or another connected device). It guarantees codes and data loaded inside (e.g., payment tokens) are protected with respect to confidentiality and integrity. The definition has been derived from [92].
Token assurance level	A value that allows the TSP to indicate the confidence level of the payment token for PAN/Cardholder binding based on factors such as the token location, and more. This confidence level can be increase with additional meta data.
Token Requestor	An entity such as a mobile wallet that seeks to utilise tokenisation and initiate requests that PANs be tokenised by submitting requests to the token service provider.
Token Domain	Restricts use of a token to the specific domain for which is was intended. For example, one token domain may be a specific card-on-file merchant, while another may be for chip card transactions with an accompanying cryptogram. The definition has been derived from EMVCo [83].
Token vault	A repository, implemented by a tokenisation system that maintains the established payment token to PAN mapping. This repository is referred to as the token vault.
Token/Payment Cryptogram	A cryptogram is a unique value generated using the payment token, keys, and additional transaction data to create a transaction specific value.
TSM	The trusted service manager is the party that issues the payment application and deploys data elements to the consumer. TSM is also responsible for managing the payment application that is stored in the SE.
White-box cryptography	A method that prevents the cryptographic key from being retrieved, even if the original source code is available and could be used to hide payment credentials in a host card emulation application.

Table 10:1 Definitions and Abbreviations

10.1 Components and Tools Used in this Work

During this project work, the following were used:

ID	Device Software	Brand	Version	Description & Purpose of Use	Reference
1	Cardholder	Swiss	1964	Marcel Fehr	n/a
2	Issuer	Cornercard	n/a	Swiss card issuer supporting Apple Pay	[71]
3	Smartphone	Apple	iPhone6 10.1.1	iPhone generation supporting NFC contactless payment	[35]
	Smartphone	Samsung	Galaxy Note3 Android 5.0 KNOX 2.3	Used to install credit card reader NFC for test purposes	n/a
4	WLAN	n/a	n/a	Wireless router providing WLAN connectivity	n/a
5	Web Proxy	McAfee	7.6.2	McAfee web proxy—commercial grade SSL intercepting proxy	[70]
6	MacBook Pro	Apple	10.12.3 HW Mid 2013	HW and software would support connectivity for DSRP via iPhone as payment device	[35]
7	Virtualisation	VMware	Professional 8.5.3	VMware provided the platform to install the web proxy, windows XP, windows 2012, and KALI Linux	[72]
8	PoS reader	ACS	ACR123 SDK	Intelligent contactless reader ACR123 USB including SDK and sample programs to become familiar with the operation of a contactless PoS interface	[75]
9	Windows 2012	Microsoft	2012	Used to install ACR123 compiled software to verify contactless payment and Apple Pay compatibility	n/a
10	Windows XP	Microsoft	2003	Used to install ACR123 SDK	n/a
11	Packet capture	Wireshark	2.0.10	Used to conduct packet tracing during enrolment to verify that traffic is not bypassing proxy	[73]
12	EMV tools	Eftlab	BP Tools 16.11	EMV tools to illustrate and understand EMV specifics. BP-Tools is a set of freeware applications for everyday EFT payment transaction service development	[74]
13	Credit Card Reader NFC (EMV)	Julien MILLAU	NFC Reader Pro	Used to verify what can be retrieved from a contactless EMV credit card, and what can be retrieved from an Apple Pay card in comparison with a standard contactless credit card	[76]
14	Linux Penetration Distribution	Kali Linux	2016-02	Kali Linux offers different tools. Pursuite was used first, and then replaced with McAfee Web proxy. Other tools used were MALTEGO and DNS, to ease the graphical representation of Apple Pay services	[77]

Table 10:2 Components and Tools

10.2 Network Analysis - Screenshots

Below are some screenshots taken during the enrolment process.

10.2.1 Http Trace – Web Proxy

The following shows the http trace on the iPhone during card enrolment:

```

↓ 22:05:36 https://sp.itunes.apple.com
↓ 22:05:36 http://a1.mzstatic.com/eu/r30/Purple49/v4/34/fd/d8/34fdd8d0-ea
↓ 22:05:35 https://xp.apple.com
↓ 22:05:35 https://play.itunes.apple.com
↓ 22:05:34 https://init.itunes.apple.com
↓ 22:05:26 https://tds.mdes.mastercard.com
↓ 22:05:22 https://nc-pod1-smp-device.apple.com
↓ 22:05:05 https://sp.itunes.apple.com
↓ 22:05:04 https://xp.apple.com
↓ 22:05:04 https://play.itunes.apple.com
↓ 22:05:03 https://init.itunes.apple.com
↓ 22:04:54 https://nc-pod1-smp-device.apple.com
↓ 22:04:54 https://tds.mdes.mastercard.com
↓ 22:04:52 https://nc-pod1-smp-device.apple.com
↓ 22:04:50 https://nc-pod1-smp-device.apple.com
↓ 22:04:35 https://nc-pod1-smp-device.apple.com
↓ 22:04:35 https://nc-pod1-smp-device-asset.apple.com
↓ 22:04:27 https://p23-fmfmobile.icloud.com
↓ 22:04:26 https://xp.apple.com
↓ 22:04:25 https://init.itunes.apple.com
↓ 22:04:24 https://play.itunes.apple.com
↓ 22:04:23 https://init.itunes.apple.com
↓ 22:04:23 https://init.itunes.apple.com
↓ 22:04:17 https://nc-pod1-smp-device.apple.com
↓ 22:04:05 https://p23-keyvalueservice.icloud.com
↓ 22:04:05 https://p23-keyvalueservice.icloud.com
↓ 22:04:04 https://pr-pod1-smp-device-asset.apple.com
↓ 22:03:59 https://p23-keyvalueservice.icloud.com
↓ 22:03:58 https://p23-keyvalueservice.icloud.com
↓ 22:03:56 https://configuration.apple.com
↓ 22:03:56 https://configuration.apple.com
↓ 22:03:56 https://p23-fmfmobile.icloud.com
↓ 22:03:55 https://init.itunes.apple.com
↓ 22:03:54 https://gsp10-ssl.apple.com
↓ 22:03:54 https://gsp10-ssl.apple.com
↓ 22:03:53 https://xp.apple.com
↓ 22:03:53 https://play.itunes.apple.com

↓ 22:03:53 https://init.itunes.apple.com
↓ 22:03:53 https://init.itunes.apple.com
↓ 22:03:49 https://nc-pod1-smp-device.apple.com
↓ 22:03:49 https://nc-pod1-smp-device.apple.com
↓ 22:03:47 https://nc-pod1-smp-device.apple.com
↓ 22:03:46 https://nc-pod1-smp-device.apple.com
↓ 22:03:46 http://ocsp.apple.com/ocsp03-wwdr02/ME4wTKADAgEAMEUwQzBBw
↓ 22:03:33 https://nc-pod1-smp-device-asset.apple.com
↓ 22:03:33 https://pr-pod1-smp-device-asset.apple.com
↓ 22:03:31 https://nc-pod1-smp-device.apple.com
↓ 22:03:30 https://nc-pod1-smp-device.apple.com
↓ 22:02:15 https://nc-pod1-smp-device.apple.com
↓ 22:01:45 https://pr-pod1-smp-device-asset.apple.com
↓ 22:01:44 https://pr-pod1-smp-device-asset.apple.com
↓ 22:01:43 https://nc-pod1-smp-device.apple.com
↓ 22:01:27 https://nc-pod1-smp-device.apple.com
↓ 22:00:56 https://nc-pod1-smp-device.apple.com
↓ 22:00:56 http://ocsp.apple.com/ocsp04-apple-root-cag3/ME4wTKADAgEAMEUw
↓ 22:00:56 http://ocsp.apple.com/ocsp04-applesica301/ME4wTKADAgEAMEUwC
↓ 22:00:50 https://nc-pod1-smp-device.apple.com
↓ 22:00:49 https://gsa.apple.com
↓ 22:00:48 https://gsa.apple.com
↓ 22:00:33 https://gsa.apple.com
↓ 22:00:20 https://p23-keyvalueservice.icloud.com
↓ 22:00:19 https://p23-keyvalueservice.icloud.com
↓ 22:00:19 https://p23-keyvalueservice.icloud.com
↓ 22:00:19 https://p23-keyvalueservice.icloud.com
↓ 22:00:19 https://keyvalueservice.icloud.com
↓ 22:00:18 https://keyvalueservice.icloud.com
↓ 22:00:18 https://gsa.apple.com
↓ 22:00:09 https://configuration.apple.com
↓ 22:00:09 https://configuration.apple.com
↓ 22:00:02 https://gsa.apple.com
↓ 22:00:01 https://gsa.apple.com
↓ 22:00:00 https://nc-pod1-smp-device.apple.com

```

Figure 10:1 Network Analysis–HTTP Trace

10.2.2 DNS Resolution of Services

Using the `#dig` command, we obtained the following output for the services accessed. This shows that most services are accessed via an AKAMAI-controlled infrastructure.

```

chrrfm-mac:~ marcelfehr$ dig -f applePayEnrolment.txt

tds.mdes.mastercard.com. 1028 IN      A          216.119.218.153
al.mzstatic.com.        2264 IN      CNAME     al.mzstatic.itunes-apple.com.akadns.net.
al.mzstatic.itunes-apple.com.akadns.net. 2213 IN CNAME     al.mzstatic.com.edgesuite.net.
configuration.apple.com. 80983 IN    CNAME     configuration.apple.com.edgekey.net.
configuration.apple.com.edgekey.net. 84 IN CNAME     e5153.e9.akamaiedge.net.
gsa.apple.com.          79324 IN    CNAME     gsa.apple.com.akadns.net.
gsa.apple.com.akadns.net. 132 IN     A          17.171.74.166
gsas.apple.com.         80995 IN    CNAME     gsas.apple.com.akadns.net.
gsas.apple.com.akadns.net. 216 IN    A          17.141.5.97
gsp10-ssl.apple.com.    356 IN     CNAME     gsp10-ssl.ls-apple.com.akadns.net.
gsp10-ssl.ls-apple.com.akadns.net. 256 IN A          17.167.193.162
init.itunes.apple.com.  660 IN     CNAME     init-cdn.itunes-apple.com.akadns.net.
init-cdn.itunes-apple.com.akadns.net. 660 IN CNAME     itunes.apple.com.edgekey.net.
keyvalueservice.icloud.com. 79309 IN  CNAME     keyvalueservice.fe.apple-dns.net.
keyvalueservice.fe.apple-dns.net. 10 IN A          17.248.146.110
nc-pod1-smp-device-asset.apple.com. 71829 IN CNAME     smp-device-content.apple.com.edgekey.net.
smp-device-content.apple.com.edgekey.net. 15478 IN CNAME     e9959.e9.akamaiedge.net.
nc-pod1-smp-device.apple.com. 1371 IN  CNAME     nc-pod1-smp-device.gcsis-apple.com.akadns.net.
nc-pod1-smp-device.gcsis-apple.com.akadns.net. 60 IN A          17.171.78.6
ocsp.apple.com.         103 IN     CNAME     ocsp.pki-apple.com.akadns.net.
ocsp.pki-apple.com.akadns.net. 37 IN     A          17.171.8.16
ocsp.pki-apple.com.akadns.net. 37 IN     A          17.171.8.16
p23-fmfmobile.icloud.com. 81327 IN  CNAME     p23-fmfmobile-current.edge.icloud.apple-dns.net.
p23-fmfmobile-current.edge.icloud.apple-dns.net. 30 IN A          17.248.146.181
p23-keyvalueservice.icloud.com. 79678 IN CNAME     p23-keyvalueservice-current.edge.icloud.apple-dns.net.
p23-keyvalueservice-current.edge.icloud.apple-dns.net. 4 IN A          17.248.146.175
play.itunes.apple.com.  1168 IN    CNAME     play-cdn.itunes-apple.com.akadns.net.
play-cdn.itunes-apple.com.akadns.net. 1168 IN CNAME     itunes.apple.com.edgekey.net.
pr-pod1-smp-device.apple.com. 1722 IN  CNAME     pr-pod1-smp-device.gcsis-apple.com.akadns.net.
pr-pod1-smp-device.gcsis-apple.com.akadns.net. 60 IN A          17.141.128.6
sp.itunes.apple.com.    3491 IN    CNAME     sp-cdn.itunes-apple.com.akadns.net.
sp-cdn.itunes-apple.com.akadns.net. 3491 IN CNAME     sp.itunes-apple.com.akadns.net.
xp.apple.com.           2381 IN    CNAME     xp.itunes-apple.com.akadns.net.
xp.itunes-apple.com.akadns.net. 83 IN     CNAME     mt-ingestion-service-mr22.itunes.apple.com.

```

Figure 10:2 Network Analysis–DNS Output

10.2.3 Location Overview

The image below shows how the services accessed during card enrolment are globally distributed. Note that this image displays a snapshot. In the case that services are hidden behind AKAMAI DSN services (cname), the final IP address shown below might change owing to re-routing. All service IP addresses are in the US.

1. One section of Apple’s services is available through AKAMAI owned IP addresses
2. Master Card Enablement services are separately hosted
3. Other Apple services are available through apple owned IP addresses



Figure 10:3 Network Analysis–Global Service Distribution

10.3 Various Screenshots

This section contains various screenshots created during the analysis of the wallet solutions.

10.3.1 Access to Card Data via NFC Interface

The images below show how much sensitive data can be retrieved from a contactless EMV card with a standard NFC reader application [76], obtained from Julien Millau running on Android.

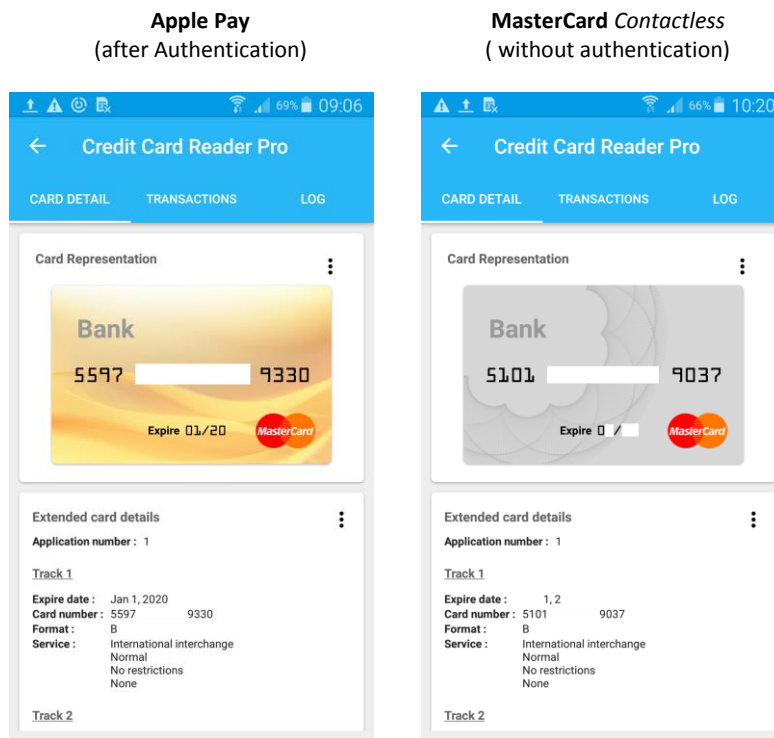


Figure 10:4 NFC Android Card Reader

10.3.2 Payment Receipts using Apple Pay at PoS Contactless

These are payment receipts from two different countries. In Australia, Apple Pay was literally supported where the contactless sign was present. In Switzerland, acceptance is far more restricted, to just a few shops.

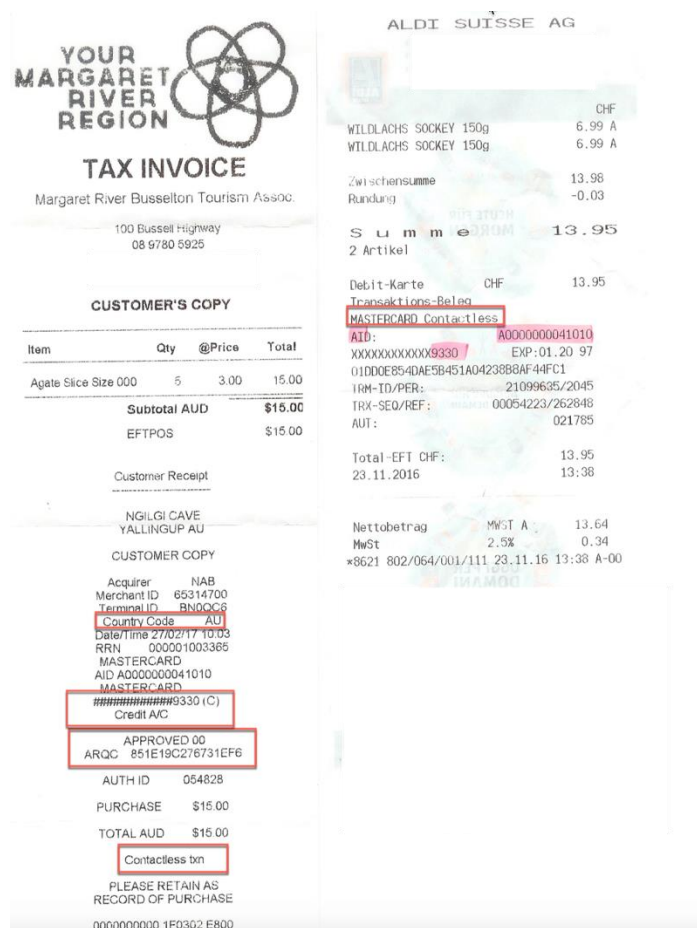


Figure 10:5 Apple Pay Receipts

10.3.3 Transaction History using Apple Pay at PoS Contactless

Apple Pay provides a transaction log, where one can view previous transactions that were approved or declined. In our case, these were declined, because sufficient funds were not preloaded. This also shows that we obtained online payment authorisation, where the issuer can reject an authorisation in its authorisation response message.

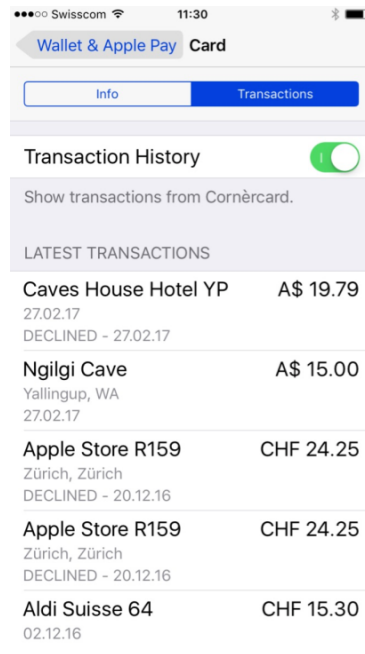


Figure 10:6 Apple Pay-Transaction History

10.3.4 Apple Pay at PoS Contactless ACR 123 Reader

For verification purposes, the author purchased an ACR contactless reader [75] that is compatible with MasterCard's *PayPass*. The provider's reader application worked from the beginning with Apple Pay and the loaded MasterCard, without requiring any changes. Below are a few screenshots from a demo payment, ordered from left to right and then top to bottom.

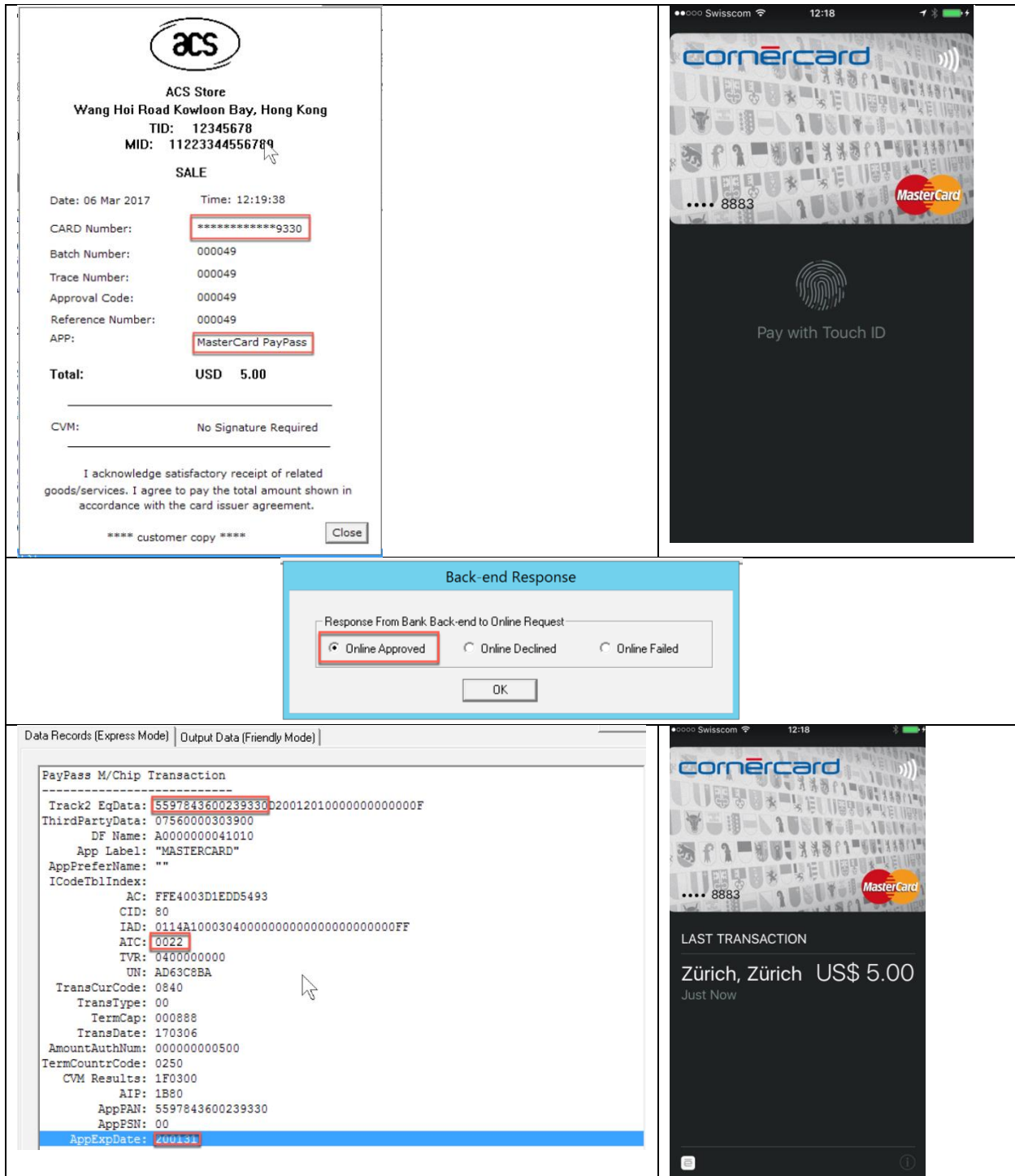



Figure 10:7 Apple Pay—ACS Reader Application Acceptance

10.3.5 Use of tokenPAN in CNP Transactions

For verification purposes, the author attempted to employ the device account number (tokenPAN) in a CNP transaction, which should not be, and was not, possible. The payment service provider cancelled the transaction, and the web interface returned an error message.



Zahlung nicht erfolgreich

Ein unbekannter Fehler aufgetreten, und Ihre Zahlung konnte nicht abgeschlossen werden.

Referenznummer: 7293836. Bitte geben Sie diese Nummer und den allfällig erhaltenen Fehlercode immer an, wenn Sie uns im Zusammenhang mit dieser Nachricht kontaktieren.

Zurück zur Kasse

Translation:
Zahlungsprozess läuft: payment process running
Zahlung nicht erfolgreich: payment failed

Figure 10:8 Apple Pay—Use of Device Account Number in a CNP Transaction

10.3.6 Apple Pay–Payment Sheet

Below is a sample payment sheet, for visualisation purposes.

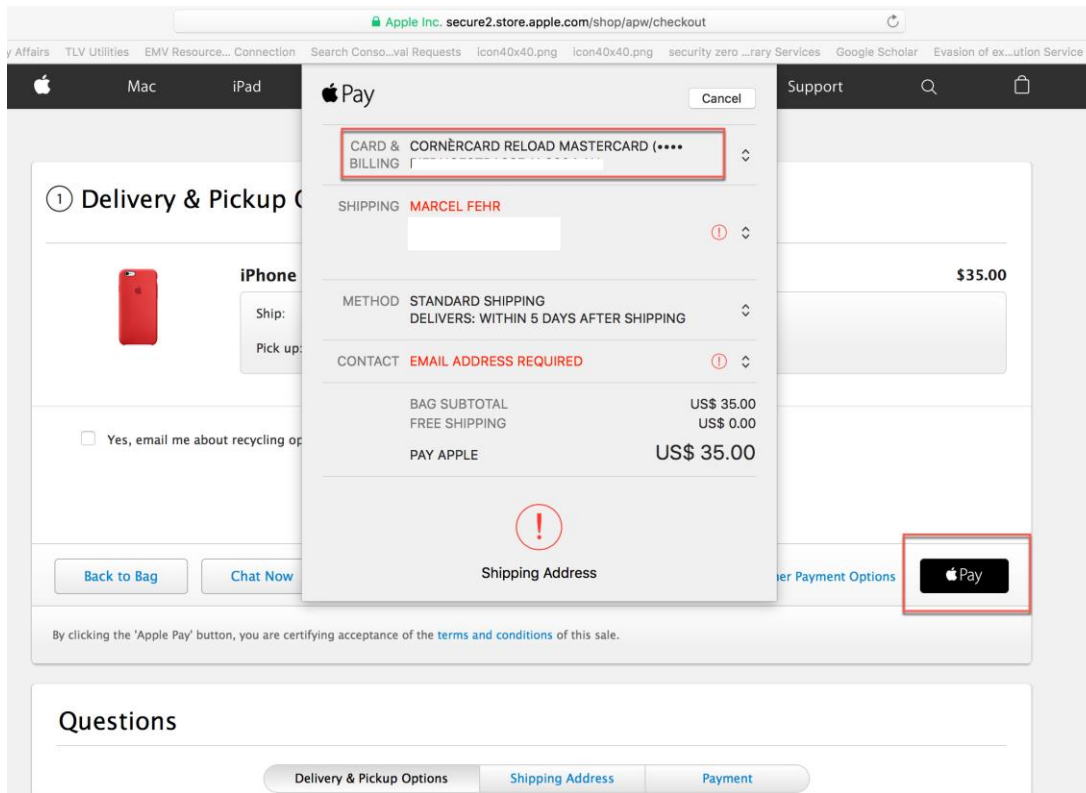


Figure 10:9 Apple Pay–Payment sheet in a CNP Transaction