Security of Mobile Payments and Digital Wallets

DECEMBER 2016

www.enisa.europa.eu

European Union Agency For Network and Information Security

About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

Contact

For queries in relation to this paper, please use info@enisa.europa.eu
For media enquires about this paper, please use press@enisa.europa.eu.

Acknowledgements

Romana Sachovà, Fraud Prevention Manager, CaixaBank Soralys

Mario Maawad Marcos, Fraud Prevention Director, CaixaBank

Soralys Hernandez Revetti, Security Consultant, GMV

Legal notice

Notice must be taken that this publication represents the views and interpretations of ENISA, unless stated otherwise. This publication should not be construed to be a legal action of ENISA or the ENISA bodies unless adopted pursuant to the Regulation (EU) No 526/2013. This publication does not necessarily represent state-of the-art and ENISA may update it from time to time.

Third-party sources are quoted as appropriate. ENISA is not responsible for the content of the external sources including external websites referenced in this publication.

This publication is intended for information purposes only. It must be accessible free of charge. Neither ENISA nor any person acting on its behalf is responsible for the use that might be made of the information contained in this publication.

Copyright Notice

978-92-9204-199-1, 10.2824/67533

Table of Contents

Executive Summary

The use of a mobile to effect payment for goods and services represents a paradigm shift towards digital only payments and has been driven by consumers who wish to make purchases at retail stores or to transfer funds using their mobile "digital wallet". For most consumers the ability to pay by mobile offers greater convenience than carrying a traditional wallet with multiple credit and debit cards.

However, using a mobile wallet is not without risks. According to a 2015 survey among mobile payment users in the US[1] "20 % affirmed their main security concern with regards to mobile payment is the possibility of someone intercepting their payment information or other data, while about 13 % feared their phones being hacked."

Furthermore, another survey[2] of more than 900 security experts concluded that only 23% of them believe that mobile payments are currently sufficiently robust at keeping personal information safe, nearly half of respondents (47%) felt that mobile payment applications offer no security and 30% of respondents were unsure.

Therefore, despite this push towards mobile payments, security concerns still remain of paramount importance and one could say that consumer discomfort with the current state of play has inhibited mass adoption.

The explosive proliferation of viruses and malware affecting mobile devices alongside the very real danger of lost or stolen devices has instilled a sense of uneasiness in the consumer mind about the implications of losing a large part of their digital lives. If we add a second dimension of money to this and the risk of unauthorised payments should a mobile device be lost, stolen or infected with malware then suddenly our mobile devices may become guardians of our financial freedom and the implications of losing our mobiles or them being susceptible to hacking or other such malfeasance skyrockets.

In this document we have identified the following key threats:

- **Mobile user threats** - installation of rogue and malware applications, phishing and social engineering
- **Mobile device threats** - unauthorized access, lost or stolen device
- **Mobile payment application and wallet threats** - reverse engineering, tampering with the payment application and the use of rootkits
- **Merchant threats -** Point of Sale (POS) malware, Man-in-the-Middle (MiTM) and replay attacks
- **Payment service providers' and Acquirers threats** - payment system compromise and data connectivity compromise
- **Payment Network Providers Threats**- token service compromise and denial of service
- **Issuers Threats** – payment authorization process compromise, token data compromise
- **Mobile Payment Applications Providers threats –** compromise of sensitive data, compromise of user profile managed in the cloud, token compromise and denial of service attacks

[1] Statista, "Mobile payment security concerns in the United States in 2015", https://www.statista.com/statistics/244322/mobile-payment-security-concerns-of-us-consumers/

[2] ISACA, "2015 Mobile Payment Security Study", http://www.isaca.org/Pages/mobile-payment-security-study.aspx?cid=pr_1110000&appeal=pr

Given that the mobile payments are still a very nascent industry without clear standards and significant industry self-regulation it is vitally important that guidelines are produced to assist mobile payment developers and mobile payment providers towards recommended security controls which if implemented would help ensure that consumers, retailers and the financial institutions that underpin the ecosystem by processing and clearing transactions are all safeguarded from cyber threats. This paper has precisely this as its primary objective and as a secondary objective to define minimum measures that should be followed by mobile payment providers in the EU, we aim to provide security recommendations for organisations wishing to provide mobile payment services within the EU.

The study also identifies a number of recommendations to mitigate the threats identified:

- Customers should follow a number of minimum security measures that should be required to securely use their application
- Mobile OS providers should ensure that their OS is regularly updated to fix any security issue identified, which may jeopardise the integrity, confidentiality or availability of the system or data
- Mobile payment application developers should provide visibility to the security measures applied to the application when offering it to the clients
- Mobile payment providers should have a reliable and accurate fraud monitoring system which reliably detects transactions outside the customer’s baseline

1. Introduction to mobile payment systems and digital wallets

We will start by analysing a threat model of a mobile payment application to obtain a high level understanding of the various types of threat that may affect mobile payment applications. We will then analyse the security features that various commercial mobile payment applications have implemented. Finally, we will discuss emerging attacks targeting mobile payment applications and countermeasures that application developers.

Digital wallets

They store value in digital form and allow an individual to purchase an item online or send funds to friends or family. Depending on the type of digital wallet used, the information stored might include debit, credit, prepaid or loyalty card data as well as personal information of the card holder such as driver's license, health card, loyalty card(s) and other ID documents.

Mobile Wallets

Some wallets, such as Android Pay, Apple Pay, and Samsung Pay, are specific to the particular combination of software and hardware on certain devices and all seek to replace the use of traditional credit/debit cards with mobile phones.

Digital currency wallets

They work in a different manner than traditional digital wallets. They typically store private keys representing ownership of a digital currency, such as Bitcoin. Once a user wants to transfer value to another user thereby paying for a good a service or simply remitting funds, then the private key is used to sign over ownership of that digital asset to the second user. The wallet then broadcasts the transaction to a network of clients who race amongst themselves to verify the transaction and include it within the distributed ledger, also known as Blockchain[3].
As soon as the transaction is confirmed on the Blockchain, then the payment is said to have happened. Digital currency wallets will not be a focus of this paper.

Contactless Payment Communication Technologies

Device based mobile wallets can use different types of communications technologies for transmitting payment data from the mobile payment device to the merchant Point of Sale (POS). Some forms of mobile to POS communication include Magnetic Secure Transmission[4] (MST), Near Field Communication[5] (NFC),

[3] Ali, Robleh, et al. "Innovations in payment technologies and the emergence of digital currencies." Bank of England Quarterly Bulletin (2014): Q3.
[4] What is MST (Magnetic Secure Transmission)? http://www.samsung.com/us/support/answer/ANS00043865/
[5] Want, Roy. "Near field communication." IEEE Pervasive Computing 3.10 (2011): 4-7.

Quick Recognition (QR) Code[6], Bluetooth[7], Bluetooth Low Energy[8] (BLE), and short message service[9] (SMS), as well as the Internet.

Mobile Payment Transactions

Mobile payments that are processed through credit and debit card networks do not change the fundamental design of the system that is already set up for traditional card based card payments.
In order to conduct a traditional card based payment transaction typically the cardholder initiates the transaction by transmitting payment authorization data, including the primary account number (PAN) to the merchant such as by swiping the card at a point of sale (POS) terminal or by inserting the EMV chip card at the POS terminal. The merchant then relays the information to the acquirer bank (the merchant bank) and then the card network relays this to the bank issuer for the payment to be authorised.
The exact same process is replicated when performed via a mobile device and contactless POS terminal, with the sole difference that the card number (PAN) and the CVC (card verification code) are typically substituted with what are called tokens instead of the actual PAN and CVC. The reason this is performed is to prevent the actual card number being sent over the wire and subsequently stored in intermediary servers.
The tokens are typically generated by the card issuers themselves who are the Token Service Providers (TSP). TSPs maintain the mapping of tokens to the corresponding PAN in a token lookup table stored in a secure database, the token vault. When a token is presented to the card issuer, the card issuer references the token lookup table and retrieves the real PAN that corresponds to that token which is then subsequently used in downstream authorisation processes.

Use of tokens and cryptograms to authorize mobile payment transactions

The security of the token and the cryptogram (a one-time encrypted string representing transaction and merchant information) are fundamental to the overall security of the mobile payment transaction itself. How the tokens are handled by the mobile payment app, such as the security of token in storage and in transit, as well as the design of the mobile application are key security considerations.

The Secure Element

The Secure Element (SE) is a tamper resistant chip with a secure microcontroller which is designed to securely store confidential and cryptographic data. The SE is a critical component in every mobile payment application but the way it is used varies greatly depending on the type of mobile payment application and also the type of mobile payment modes used; we will discuss the implementation details in the section that follows.

The Trusted Execution Environment

The Trusted Execution Environment (TEE) is a compartmentalized trusted and secure execution environment that offers a higher level of functionality than the Secure Element alone. A whole application

[6] Walsh, Andrew. "Quick response codes and libraries." Library Hi Tech News 26.5/6 (2009): 7-9.
[7] Haartsen, Jaap C. "The Bluetooth radio system." IEEE personal communications 7.1 (2000): 28-36.
[8] Gomez, Carles, Joaquim Oller, and Josep Paradells. "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology." Sensors 12.9 (2012): 11734-11753.
[9] Ayabe, Benson S., Sharat Subramaniyam Chander, and Semyon B. Mizikovsky. "Short message service." U.S. Patent No. 6,141,550. 31 Oct. 2000.

can run within the TEE and use the SE and other phone functionality but yet operate with a high degree of assurance of its confidentiality and integrity. In a mobile application, the code running on a TEE can execute sensitive functions such as storing and matching user identifiable fingerprint data.

2. Mobile Payments Platforms and Key Security Features

We will now show the most popular mobile payment/digital wallet applications: Apple Pay, Android Pay/Google Wallet and Samsung Pay. The focus of this analysis is to highlight (not compare) the security features incorporated by design in each of these mobile payment applications. There are other mobile payment applications which we haven't reviewed, such as Microsoft wallet, Paypal and others, but for the purpose of this document we will focus on the three abovementioned platforms.

2.1 Apple Pay

Apple Pay is Apple's solution for mobile payments using iOS devices including the Apple Watch. It is designed to protect cardholder personal information and allows the user to execute payments with merchants that have deployed point of sales terminals that support Apple Pay contactless payments.

Apple Pay combines a number of existing security technologies and security controls which allow users to initiate payments and to authorize payment transactions between users, merchants and card issuers.

2.1.1 Card enrolment

The first step to using Apple Pay involves the process of adding a new debit or credit card. The following diagram shows the process from a high level perspective:

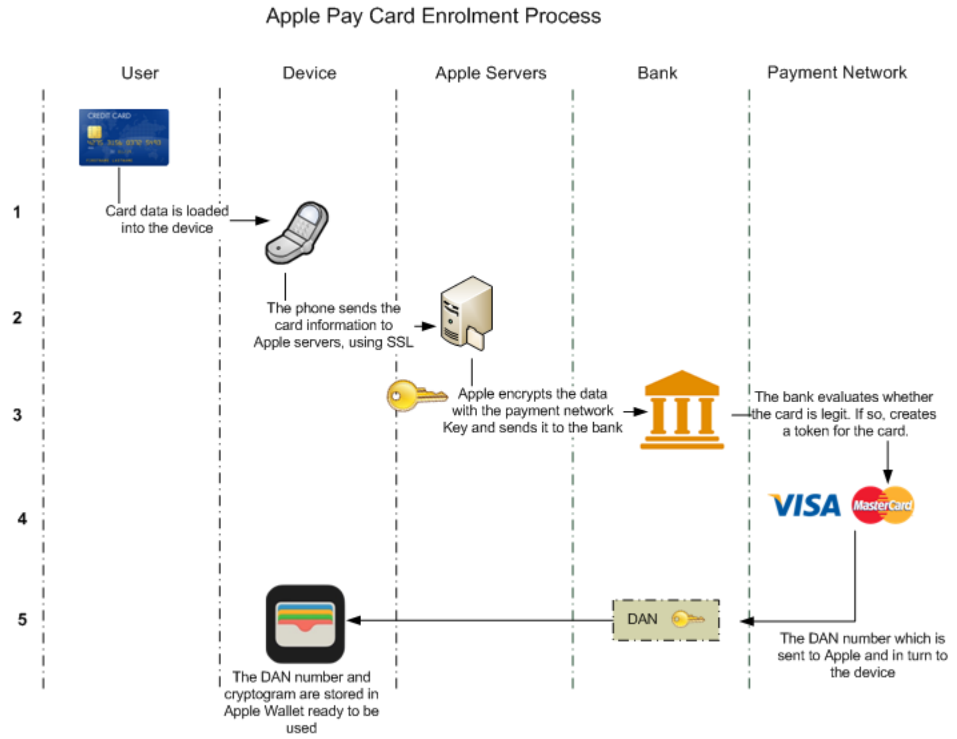
Apple Pay Card Enrolment Process
User
Device
Apple Servers
Bank
Payment Network
1
2
3
4
5
Card data is loaded into the device
The phone sends the card information to Apple servers, using SSL
Apple encrypts the data with the payment network Key and sends it to the bank
The bank evaluates whether the card is legit. If so, creates a token for the card.
VISA
MasterCard
DAN
The DAN number which is sent to Apple and in turn to the device
The DAN number and cryptogram are stored in Apple Wallet ready to be used

Figure:1 Apple Pay Card Enrolment Process

1. The user adds their existing card information by typing it into the app or by taking a picture of the credit or debit card with their phone.

2. The device receives the card information and sends it over a secure connection to Apple servers, along with other user and device data, such as iTunes/App Store activity, device information (mobile number, model, etc.) and user location (if Location Services are enabled)
3. The bank receives the information and makes a decision on whether the card is valid or not, usually interacting with the payment network provider. The information received (mobile number, model, PAN, etc.) will be used in a risk management process to determine whether the request is legitimate and the card really belongs to the user. This is particularly relevant to minimize fraud, as the same card can be enrolled in a number of devices.
4. If the card is accepted, the bank communicates with the payment network provider to create a unique token. The token is normally created by a TSP (Token Service Provider) rather than the bank itself.
5. A DAN (Device Account Number) specific for that card and device will be generated by the TSP. It will then be sent back to Apple servers along with a cryptogram which will be used to generate security codes during payment. This data will be relayed to the device, which will store it in the Passbook / Apple Wallet for future use.

2.1.2 Payment Process

The following diagram shows Apple Pay payment process from a high level perspective.

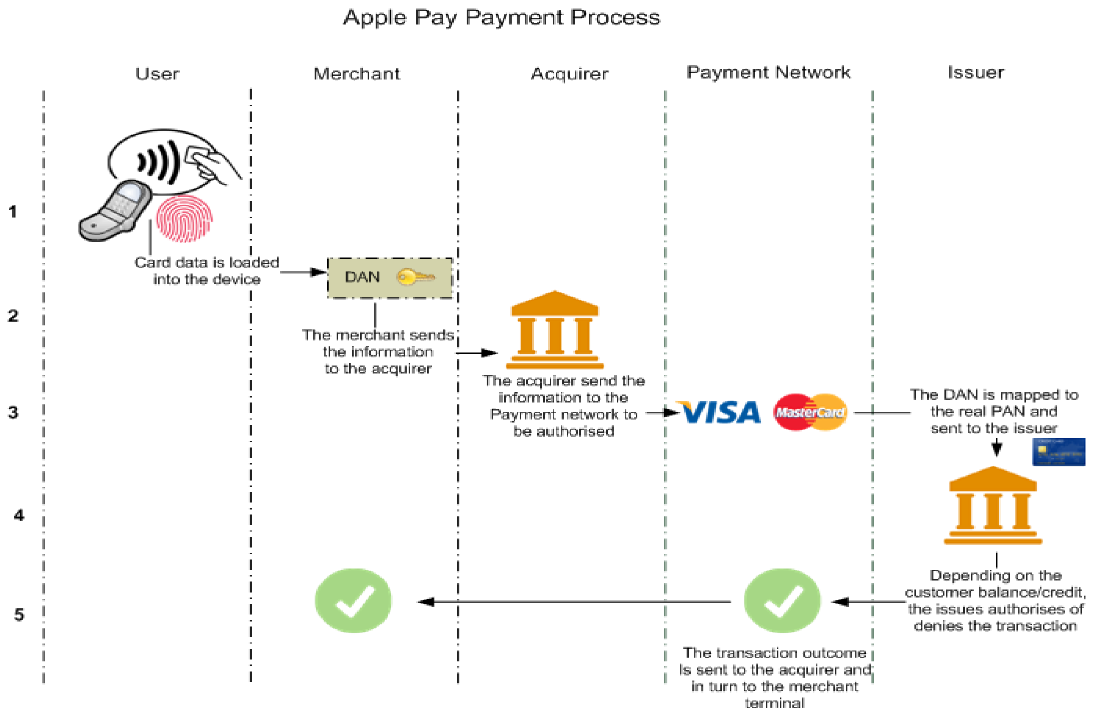
Apple Pay Payment Process
User
Merchant
Acquirer
Payment Network
Issuer
1
2
3
4
5
Card data is loaded into the device
DAN
The merchant sends the information to the acquirer
The acquirer send the information to the Payment network to be authorised
VISA
MasterCard
The DAN is mapped to the real PAN and sent to the issuer
Depending on the customer balance/credit, the issues authorises of denies the transaction
The transaction outcome Is sent to the acquirer and in turn to the merchant terminal

Figure:2 Apple Pay Payment Process

1. To initiate the payment process, the user places their device close to the NFC payment terminal. Apple Pay relies on TouchID (or a PIN number in the case of Apple Watch) to identify the user. Once a card is chosen, its token (DAN number) is loaded into the SE (Secure Element). Apple supports EMV Contactless and therefore, if it is also supported by the terminal, the SE will generate a dynamic cryptogram.
2. The merchant sends the information to the acquirer. The acquirer is the bank which will get paid for the credit card transaction.

3. The acquirer receives the DAN number, but it is unaware whether it is a valid PAN or a token. In fact, the acquirer simply verifies the BIN (Bank Identification Number) and sends it to the appropriate issuer via the payment network, which acts as an intermediary between the acquirer and the issuer.
4. The payment network will detect that it is actually a DAN instead of a real PAN, and therefore will forward the number to the TSP (Token Service Provider) to send the real PAN back to the issuer.
5. The issuer will authorise or deny the transaction and will send the notification to the acquirer, which will in turn send it back to the merchant

2.1.3 User Authentication

Apple Pay requires the user to authenticate to the device in order to perform a payment. The authentication is performed by fingerprint identification sensor (the TouchID) or a PIN number in an Apple Watch. The aim of this security control is to limit what an attacker can do with a stolen device. The use of fingerprint identification or authentication to initiate a payment is a step forward in security, compared to traditional contactless payment where a stolen card could be used without any user identification/ authentication however it is not without its risks as discussed previously (e.g., multiple enrolments, fingerprint bypass).

2.1.4 Device Authentication

Each Apple Pay transaction produces a unique value that ensures that the transaction is coming from an authorized device. This unique identifier along with the token and the cryptogram used to authorize the transaction ensure that even if the token is stolen it can't be used from another device because the token must come from the device to which it was registered. Additionally, the token is calculated with the transaction amount, and therefore even if it was intercepted in transit, it could not be used by an attacker to perform another purchase.

2.1.5 Data Protection

Apple Pay enforces data security by design, with the following controls:

1. Tokenization: During card enrolment a token is created, which is stored on the device and used during payment operations. During payment, the real PAN and card verification (CVV) numbers are never used. This design decision minimizes the exposure of real confidential data and allows the user to quickly block a card if the device has been stolen, having the card working. This approach also limits attacks from untrusted merchants, who never have visibility of the real PAN or CVV.
2. Leveraging the Secure Element: The Secure Element (SE) present in Apple devices is a highly secure chip that is tamper proof e.g., should it detect any attempts at reading its contents, it automatically zeros memory ensuring that no keys can be extracted.
3. Credit or debit card data is sent from the payment network or card issuer encrypted using payment applets that reside in the secure element.
4. During a transaction, the terminal communicates directly with the Secure Element through the Near Field Communication (NFC) controller over a dedicated hardware bus.
5. Payment authorization details for contactless transactions are localised to the local NFC field and are never exposed to the application processor

2.2 Google Wallet/Android Pay

Initially Google Wallet relied on the Secure Element as its trusted store of sensitive payment information. Google has since changed direction, using Host Card Emulation after announcing Android Pay in May 2015, which in effect means that payment credentials are stored in the cloud.

Apart from the transition from SE to HCE, Google also changed the authentication process, added loyalty rewards and integrates with other apps.

These changes brought significant architectural changes to the solution, which in turn had an impact on the attack surface.

2.2.1 Card enrolment

Users of Android Pay must first register their debit or credit cards with Android Pay. Android Pay (and Google) offloads the liability of identifying the user to the customer's bank. Therefore, it just provides a number of identification options, which can be used by the card issuer to decide whether the customer identity is verified. The following ways of verification are offered:

1. Verifying by email or text: The customer's bank will send the customer an email/text with a verification code.
2. Verifying by phone: The customer could call the bank and request the verification code.
3. Verification via the bank's app: If the customer has the bank's application already installed on the mobile, it is possible to sign in to the app to verify the card.
4. Verifying with "temporary charge": This verification process will charge the user's account with a very small charge, including a 6 digits code. The user would need to log on to the electronic banking and provide the verification code.

The user enrolling a card on Android Pay needs to be aware that the card number is going to be transmitted and stored in Google's cloud server.

2.2.2 Payment Process

Google Wallet was initially implemented using a Secure Element based model to securely store encrypted sensitive data such as cardholder data and card verification codes in the form of tokens on the actual device itself device.

A decision was made in 2014 to consider the device untrusted and compromised and instead move the SE to the cloud using HCE, this doesn't mean that the SE was itself thought to have been a compromised component.

The following diagram is a high level representation of Android Pay payment.

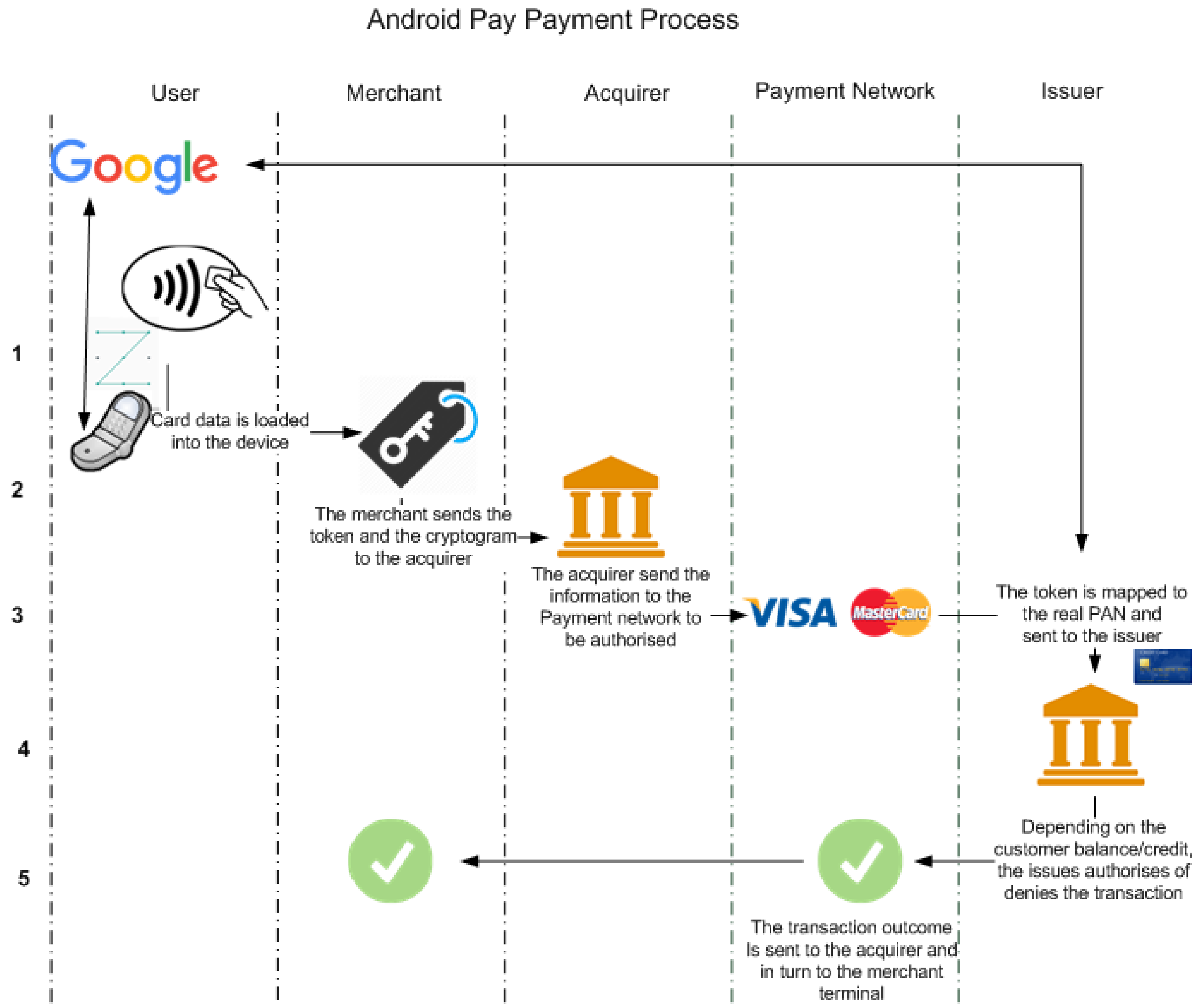
Android Pay Payment Process
User
Merchant
Acquirer
Payment Network
Issuer
Google
1
2
3
4
5
Card data is loaded into the device
The merchant sends the token and the cryptogram to the acquirer
The acquirer send the information to the Payment network to be authorised
VISA
MasterCard
The token is mapped to the real PAN and sent to the issuer
Depending on the customer balance/credit, the issues authorises of denies the transaction
The transaction outcome Is sent to the acquirer and in turn to the merchant terminal

Figure:3 Android Pay Payment Process

1. Before the payment process begins, the device has connected to Google servers and has been provided with a number of valid payment tokens. When the user places the device close to the NFC POS, HCE enables the NFC controller on the device, which will handle the communication between the POS and the wallet, requesting one of the tokens. The dynamic token and the cryptogram are sent to the POS.
2. The merchant sends the information to the acquirer. The acquirer is the bank which will get paid for the credit card transaction.
3. The acquirer receives the token and the cryptogram and sends it to the appropriate issuer via the payment network, which acts as an intermediary between the acquirer and the issuer.
4. The payment network will request the real PAN from the TSP (Token Service Provider) and send it to the issuer for approval.
5. The issuer will authorise or deny the transaction and will send the notification to the acquirer, which will in turn send it back to the merchant.

2.2.3 User Authentication

Android Pay offers a number of options to authenticate the user before payment. Android Pay accepts fingerprint authentication (not enabled by default), PIN code, password, or pattern to authenticate a transaction.

Whereas in traditional card payments the user tends to protect the PIN number (which authenticates the user), mobile patterns are commonly displayed in public and may introduce a significant threat to Android Pay security model.

2.2.4 Device Authentication

Payment tokens are loaded on the device in advance, before the payment. Tokens are periodically retrieved from Google servers when connectivity is available.

As an additional measure, Android Pay is designed not to run on devices, which have administrative (superuser) control enabled (also known as root access).

2.2.5 Data Protection

Since HCE assumes that any data stored on a handset is vulnerable (e.g. in case of a stolen device or compromised device by malware) it stores the card sensitive data on databases hosted in a secure cloud environment.

Preventing unauthorized access to the HCE depends on four security pillars:

1. limited use security keys,
2. tokenization,
3. device fingerprinting, and
4. transaction risk analysis.

Limited use keys expire quickly preventing their misuse. Tokens reduce risk by replacing the PAN with limited use data that passes seamlessly through the payment system. Device profiles (fingerprints) can validate the phone. Data analysis provides real-time transaction assessment to identify unusual activity[10].

2.3 Samsung Pay

Samsung Pay is a mobile wallet application that leverages MST (Magnetic Secure Transmission) for making payments by emulating a magnetic card stripe reader. Tokens are sent to the Point of Sale terminal after the user authenticates to the Samsung Pay App. Use of MST enables Samsung Pay provisioned phones to make payment not only at NFC tap-and-pay devices but also at traditional magnetic stripe terminals hence extending their reach much wider unlike Apple and Android pay which require both retail and bank support. Samsung Pay is closely coupled to the Samsung KNOX platform, which provides functionality for encrypted storage of payment tokens.

[10] Smart Card Alliance, Mobile & NFC Council, “ Host Card Emulation (HCE) 101”, http://www.smartcardalliance.org/downloads/HCE-101-WP-FINAL-081114-clean.pdf

2.3.1 Card Enrolment

Samsung Pay allows customers to add bank cards (credit cards, debit cards, and store credit cards) and gift cards.

During the enrolment process, the user has to add the card to Samsung Pay as well as performing an identity validation. The user is provided with SMS, Email and Call Bank as possible authentication options to send a One Time Password (OTP). In comparison with Android Pay (which also offers authentication via the issuer's banking application and "temporary change"), Samsung's offering is limited.

After adding the card to Samsung Pay, there is a verification process performed by the payment card network (i.e., Visa, MasterCard, or American Express) and the card issuer. According to Samsung, the following information is shared with the issuer:

1. Samsung Account information and Samsung Pay usage data, such as usage duration and how many cards the user may have registered on Samsung Pay, as applicable;
2. Device information, such as device model number, OS version and other device identifier(s);
3. Location information (i.e., where the user is during the card registration), but only if location detection is enabled on the device at the time of the registration; and
4. Card information and billing address, which is sent to the card issuer, passing through Samsung servers.

The customer cannot perform any payment with the added card until the authentication process has been successfully performed.

2.3.2 Payment Process

The following diagram shows the Samsung Pay payment process from a high level perspective

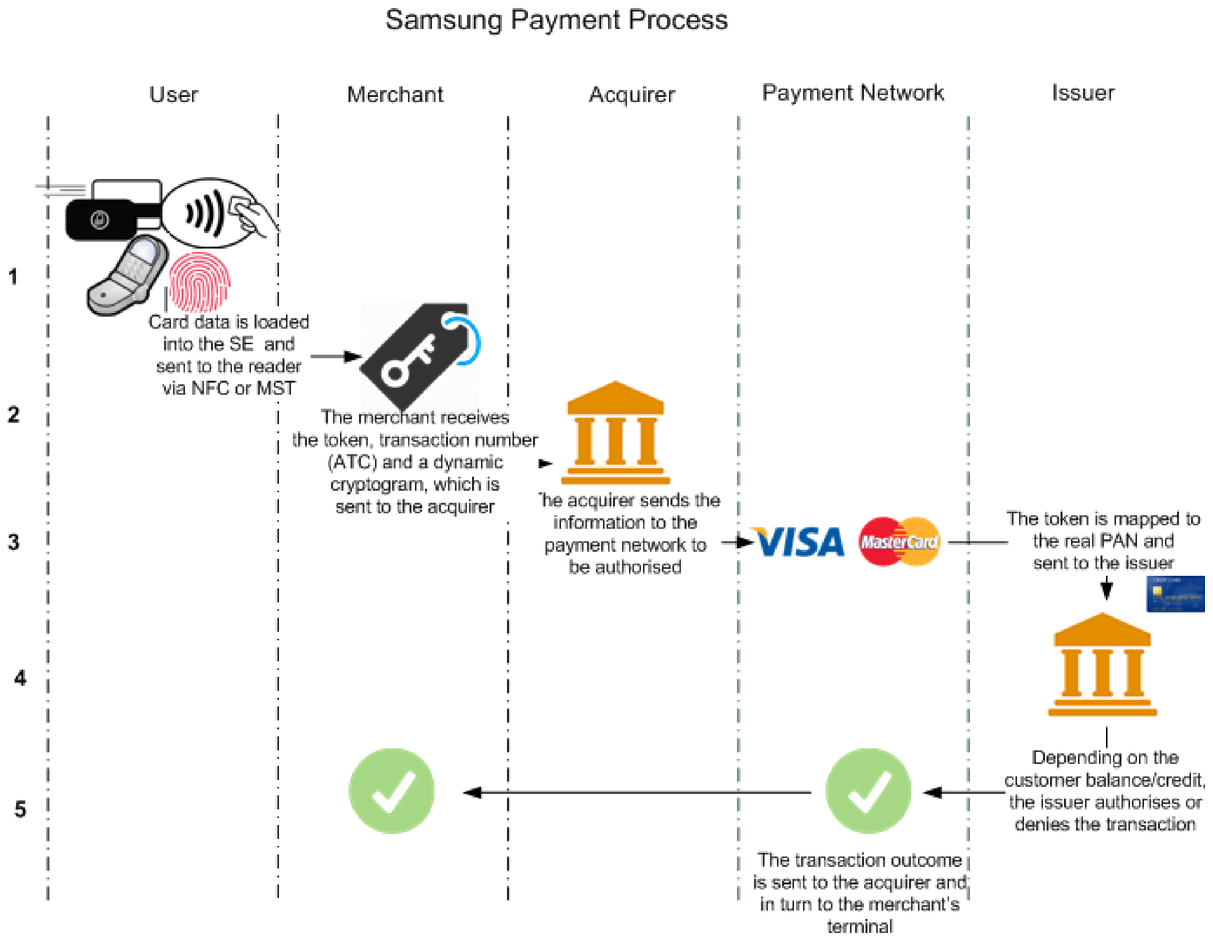
Samsung Payment Process
User
Merchant
Acquirer
Payment Network
Issuer
1
2
3
4
5
Card data is loaded into the SE and sent to the reader via NFC or MST
The merchant receives the token, transaction number (ATC) and a dynamic cryptogram, which is sent to the acquirer
The acquirer sends the information to the payment network to be authorised
VISA
MasterCard
The token is mapped to the real PAN and sent to the issuer
Depending on the customer balance/credit, the issuer authorises or denies the transaction
The transaction outcome is sent to the acquirer and in turn to the merchant's terminal

Figure:4 Samsung Pay Payment Process

The payment transaction follows these main six (6) steps:

1. The user initiates the payment by placing the handset in proximity to the NFC or magnetic stripe POS. Samsung Pay will then initiate the payment process using the NFC transmitter or MST technology. After choosing a card the handset generates 3 pieces of information:
   a. A digital token associated to the card, provisioned by the payment network. The purpose of the token is to conceal the real PAN and allow the acquirer to route transactions to the correct payment network and issuer.
   b. A transaction counter (ATC) which is incremented on each transaction and allows the payment network to keep track of payment sequence.
   c. A cryptogram generated with a secret key (which is known only to TrustZone), token and ATC.
2. The merchant reader receives the information described above and conveys the message to the acquirer.
3. The acquirer will identify the appropriate payment network and will forward the transaction information.
4. The payment network will identify the token and call the TSP to retrieve the real PAN number associated to it, which will then be forwarded to the issuer in order to execute the payment.
5. The issues will verify whether the transaction can be executed. If so, it will perform the payment.

6. The payment network will be notified of the successful transaction, which in turn will also notify the merchant.

2.3.3 User Authentication

User authentication in Samsung Pay can be performed in two ways: Fingerprint authentication or assigning a four-digit PIN. Taking into consideration that traditional credit cards do not require any additional user authentication for contactless payments, this measure provides an additional security layer.

2.3.4 Device Authentication

Samsung Pay provides a number of mechanisms for Identity and Verification (ID&V) for token assurance during token provisioning. Tokens are provided to the device in advance, when Internet connection is active. The issuer has a choice to choose the ID&V controls in use from a range which includes billing address, device ID, one time passwords (via SMS, email, call centre and app-to-app channels).

As an additional measure, Samsung Pay is designed not to run on rooted devices.

2.3.5 Data Protection

Data protection on the device relies on Samsung KNOX and the TEE (Trusted Execution Environment). Samsung KNOX provides a framework which effectively splits the hardware into two: Normal world and Secure world. Both areas are isolated and only accessible via a monitor (TrustZone monitor). TEE provides a range of hardware secure resources for key storage.

PAN numbers are not installed on the smartphone and they are only made available to the issuer for payment. Therefore, the risk of PAN leaks is minimal.

3. Potential risks for mobile payments: Threats and Vulnerabilities

3.1 Mobile Payments & Digital Wallets Threat Model

A threat model of a mobile payment application shall consider threats against basic components of the mobile application ecosystem highlighting the "trust boundaries" (depicted below as dotted red lines) that are the points of demarcation between parts of the mobile payment application where threats are most likely to occur. A generic threat model of the mobile payment ecosystem is shown below:

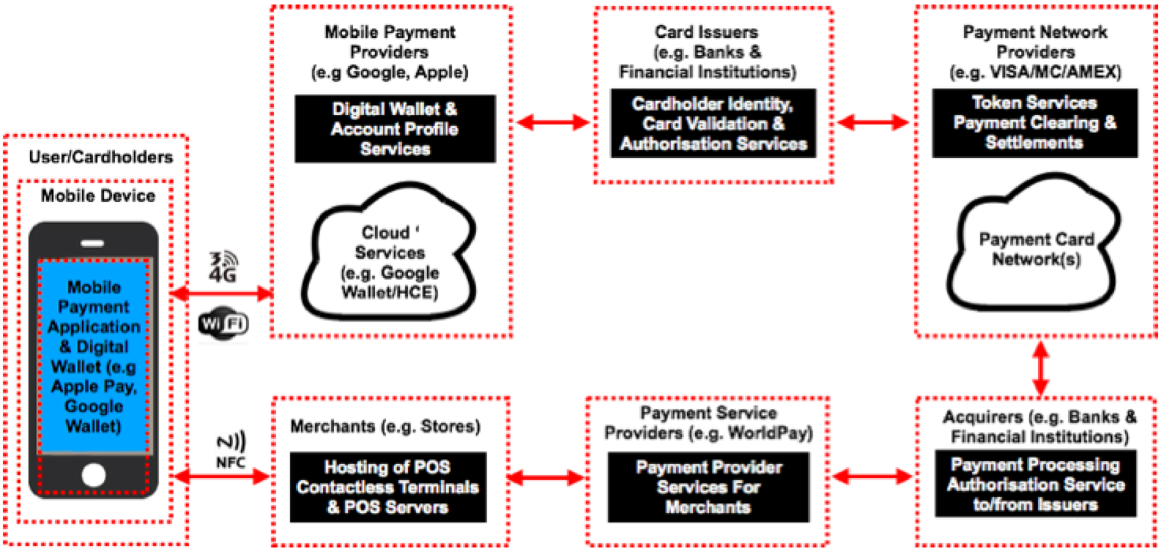
User/Cardholders
Mobile Device
Mobile Payment Application & Digital Wallet (e.g Apple Pay, Google Wallet)
3G 4G
Wi Fi
NFC
Mobile Payment Providers (e.g Google, Apple)
Digital Wallet & Account Profile Services
Cloud Services (e.g. Google Wallet/HCE)
Card Issuers (e.g. Banks & Financial Institutions)
Cardholder Identity, Card Validation & Authorisation Services
Payment Network Providers (e.g. VISA/MC/AMEX)
Token Services Payment Clearing & Settlements
Payment Card Network(s)
Merchants (e.g. Stores)
Hosting of POS Contactless Terminals & POS Servers
Payment Service Providers (e.g. WorldPay)
Payment Provider Services For Merchants
Acquirers (e.g. Banks & Financial Institutions)
Payment Processing Authorisation Service to/from Issuers

Figure:5 Mobile Payments and Digital Wallets Threat Model

We will analyse the threats and attack vectors of the main components of the mobile payment ecosystem affected by these.

3.2 Mobile Payment Application Users Threats

Threats directed against the users of mobile payment applications are:

Phishing and social engineering

Mobile phones are mixing personal and corporate usage. Mobiles are gathering more and more information from the customer, which aggregated could help to carry out sophisticated attacks.

These attacks target the user by phishing emails and social engineering exploiting different communication channels (e.g. phone, email, SMS) and data about the user available in the public domain (e.g. social media sites, search engines). The data sought by attackers using social engineering are often credit card data and personal data that the user knows about. Stolen credit/debit card or prepaid card data (e.g. PAN, CVV, card expiration date) can be either monetized (e.g. sold in underground market forums) or used for fraudulent payments. Stolen personal data of the mobile payment user (e.g. names, last name, date of birth, contact information such as billing shipping address, emails, phone numbers) can be used for impersonation attacks and for identity theft.

Installation of rogue applications and malware

Fraudsters will find ways to install malware on the mobile device by phishing/social engineering a victim to open a malicious attachment in an email and by redirecting the user to a malicious URL.

Another possible channel for malware infection is insecure WiFi hotspots (e.g. Internet cafes) that might allow an attacker to target the mobile device with Man-in-The-Middle. There is also the possibility of a network spoofing attack. That is when malicious user setups a fake access point with same network name, as one that already exists, such as popular café name or market chain. They might setup a fake website to "authenticate" users and this way collect data, then they can later use this data for next steps in their attack. It is not uncommon to see many people use same username and password for multiple different services, even for a mobile payment application.

3.3 Mobile Devices Threats

The main threats against mobile devices that host the mobile payment application are:

Unauthorized access to lost or stolen mobile device

Direct attacks assume the attacker has possession of a device that is either inadvertently lost by the user or stolen and finds its way in the hands of malicious users/attackers. Once in possession of the device, the attacker might try to access the device. Most likely attacks consist of attempts to bypass any PIN or fingerprint locks. When the device is protected via fingerprint authentication, the attacker could also use fingerprints stolen from other sources of fingerprint data e.g., lifting latent fingerprints from surfaces. An attacker in possession of the device might try to use commercial or open source forensics tools that jailbreak the device OS and gain root access to the file system to steal data installed on the device.

Malware installation on the device

The installation of malware/rootkits[11] can be facilitated by drive by download attacks leveraging e.g., WebKit to root level access, or by side-loading of malware alongside legitimate or semi legitimate apps downloaded from the various stores.

3.4 Mobile Payment & Digital Wallet Applications Threats

Reverse engineering the application source code

Often reverse engineering the binary itself is the first port of call for an attacker seeking to obtain an innate understanding of the payment application in order to exploit vulnerabilities such as hardcoded passwords and encryption keys as well as for crafting application specific attack vectors.

Tampering with the mobile payment application

An attacker may choose to backdoor a mobile payment application so as to capture login details and send these to an attacker controlled server. He would do this by downloading the legitimate application from the store, unpacking it, patching the relevant routines and then repackaging and uploading to the store. Given the proliferation of hundreds of application stores offering such applications, this is a very realistic threat on mobile devices.

[11] C.Papathanasiou, N. Percoco "This is not the droid you're looking for", https://www.defcon.org/images/defcon-18/dc-18-presentations/Trustwave-Spiderlabs/DEFCON-18-Trustwave-Spiderlabs-Android-Rootkit.pdf

Exploit of mobile payment application vulnerabilities

Exploits of mobile application vulnerabilities might allow attackers to steal any sensitive data stored by the application (e.g. personal account details of the user and credit card data). Exploit of vulnerabilities such as weak authentication might allow an attacker to gain unauthorized access to the device. Unauthorized access to mobile payment functionality might occur because of exploit of mobile payment APIs used for in-app purchases allowing an attacker to conduct fraudulent transactions. Additionally, fraud is possible with stolen bank and credit card accounts linked to the mobile payment application. A fraudster might also exploit weaknesses in the registration process to add another mobile device to the user profile to conduct fraudulent purchases.

Installation of rootkits/malware

Rootkits[12] are a significant threat vector and can also be leveraged to directly monitor and hijack / manipulate API calls as they are being marshalled to/from the mobile payment API endpoint and hence manipulate variables in transit e.g., payment amounts.

Mobile Operating System Access Permissions

A mobile OS may give access to certain resources with the permission of the user. Even if, a given application might not be malicious, holding certain permissions might potentially give access to sensitive data or be used by another application to elevate access.

3.5 Merchants Threats

Uploading malware POS on the POS contactless payment terminal

Uploading POS malware (e.g. Carbanak[13], Malum POS[14]) exploit security weaknesses at the merchant such as use of insecure remote desktop access to POS servers. Once the POS malware is installed on the POS contactless terminal it can be configured by the attacker to remotely steal payment data that transact through the card readers that might include also magnetic stripe card data and Chip & PIN EMV credit card data[15].

MiTM attacks against the POS contactless terminal and POS server connections

MiTM Attacks are possible by exploit of the following vulnerabilities:

a. the inherent lack of security of contactless communication channel used at the POS such as MST (used by Samsung Pay)

[12] Michael Davis, Sean Bodmer, and Aaron LeMasters. 2009. HACKING EXPOSED MALWARE and ROOTKITS (1 ed.). McGraw-Hill, Inc., New York, NY, USA.

[13] The Great Bank Robbery: Carbanak APT https://business.kaspersky.com/the-great-bank-robbery-carbanak-apt/3598/

[14] Trend Micro Discovers MalumPoS; Malware Targeting Hotels and other US Industries http://blog.trendmicro.com/trendlabs-security-intelligence/trend-micro-discovers-malumpos-targets-hotels-and-other-us-industries/

[15] **Note**: attacks against a mobile payment transaction token cannot be used for mobile payment impersonation fraud without also stealing the cryptogram and can only be used once in a payment transaction. This will be further discussed later in this paper. Attacks to chip and PIN card data might seek to steal PINs and magnetic stripe data (e.g. CVV, PAN) that are static data and can be used more than once.

b. SSL/TLS or end to end encryption not being used between the POS terminal (the POI) and the POS server.

Attackers can also attempt to exploit network security weaknesses such as lack of firewalls to protect the merchant internal network as well as attempt to exploit vulnerabilities in POS software and POS mis-configurations (e.g. not enforcing minimum privileges to access POS terminals and servers).

Relay attacks against NFC enabled POS contactless terminal

A known attack against the NFC POS interface is the relay attack. Relay software installed on the victim's phone can relay commands and responses between the Secure Element and a card emulator (that is installed as proxy on the mobile POS) across a wireless network. With a remote relay attack for example, android malware installed on the mobile device can allow a fraudster to conduct unauthorized payments by channelling SE communications to the remote attacker, allowing him/her to make purchases without physical possession of the targeted device[16].

3.6 Payment Service Providers Threats

Possible threats and attacks against Payment Service Providers that route mobile payments from the merchant to the merchant's acquirer banks/financial institutions are:

Payment systems compromise

Payment Service Providers (PSPs) provide POS contactless terminals for mobile payments (e.g. for NFC enabled POS terminals) as well as aggregated payment services for merchants by processing data from different channels including face to face (card present) payments, online payments and mobile/contactless payments. PSP payment gateways represent an interesting target for attackers that seek to compromise the payment data in transit from the merchants to the different acquiring banks. Attackers might seek to compromise software vulnerabilities in POS contactless terminals that PSPs provide to merchants to host on their premise/network, POS servers' software also installed at merchant POS servers and the payment gateways hosted at the payment service providers such as by exploiting un-authorized access to payment gateways and weaknesses in enforcement of internal payment service providers' security controls and measures.

Data connectivity compromise

Attackers might try to exploit insecure connections (e.g. lack of enforcement of secure connections (e.g. SSL/TLS, VPN) to conduct attacks such as MiTM to spoof sensitive data in transit from merchant hosted systems to the payment gateway hosted at the Payment Service Provider (PSP) and from the PSP to the different acquirers that the PSP routes the mobile payment data (e.g. token and cryptogram) to.

3.7 Acquirers Threats

Possible threats and attacks against acquirer's banks/financial institutions that process mobile payments on behalf of merchants are:

Payment processing systems compromise

[16] Christian Killer, Christos Tsiaras, Burkhard Stiller, University of Zürich, "An Off-the-shelf Relay Attack in a Contactless Payment Solution", https://files.ifi.uzh.ch/CSG/staff/tsiaras/Extern/Theses/VA_ChristianKiller.pdf

Since acquirers send payment authorization requests via tokens and cryptograms and receive authorizations from the issuer through the payment network, payment processing services are likely primary targets by attackers seeking to obtain large amounts of cardholder data. Attackers might seek to compromise the acquirer bank payment processing servers from the inside of the network such as by exploiting un-authorized access to payment gateways and weaknesses in enforcement of internal security controls and measures as well as remotely through installation of backdoors and Remote Access Tools (RAT) via malware infection of the servers hosted at the acquired network.

Data connectivity compromise

Attackers might try to exploit insecure point to point connections (e.g. misconfiguration, gaps and vulnerabilities in secure point to point connections) between acquirer and issuer through network service provider network to conduct attacks such as MiTM to spoof sensitive data in transit from the acquirer to/from the issuer via the payment network.

Repudiation of mobile payment authorization

Repudiation attacks such as to repudiate a payment authorization from an issuer can be facilitated by exploits of design flaws in the implementation of payment processing services by the acquirers.

For instance, not using mutual authentication of the point to point connections as well as digital signatures to validate authorization approvals and payment verification process through an independent channel from the payment network channel where these authorizations are received from.

3.8 Payment Network Providers Threats

Possible threats and attacks against payment network providers that settle mobile payments between acquirers and issuers are:

Token services provider services & servers compromise

Token Services Providers (TSP) provide token management services such as tokenization (creation of a token from credit card PAN), de-tokenization (retrieval of the PAN from a token vault) and validation of the token data integrity and origination token and validation with cryptograms. If a token service provider were compromised, attackers would likely try to obtain the token look-up tables which provide the token to PAN, CVV and expiry mappings. This would be a high value target for an attacker as it would provide them with easily useable and monetizable information.

Other possible attacks against the tokenization and de-tokenization process might involve exploit of software vulnerabilities to extract the PAN used for authorize the transactions, identification and verification of credit card data and for clearing and settlement. Attacks against the domain restrictions enforced by the TSP which may allow an attacker to bypass tokens time, place and digital channel restrictions.

Denial of payment settlement services

Attacks targeting the availability of token services hosted by payment network organization will impact the authorization of mobile payments and possibly also for payments originating from other channels (e.g. contact EMV cards channel) that also use these token services.

3.9 Issuers Threats

Possible threats and attacks against banks and financial institutions that issue cards to cardholders that are users of mobile payment applications include:

Payment authorization process compromise

One of the main threats for card issuers regard the processes that validate cardholder data and issues payment authorizations to the acquirer. An internal attacker at the card issuer bank or an external attacker that gained access to critical servers may attempt to bypass fraud controls (e.g. changing the card payment limits on authorized compromised credit cards registered for mobile payment transactions).

Confidential cardholder data compromise

Credit and debit accounts including bank account data stored at the issuer banks are highly targeted by fraudsters and cybercriminals that seek to commit fraud with stolen credit card data through counterfeit cards and card not present fraud and by reselling stolen credit card data on the black market. Even if attacks against the databases hosted at the issuer banks that store cardholder's sensitive data are unlikely because of the high security standards that are usually followed, exploits might be possible because of the following attacks:

a. Social engineering internal employees at the bank that have access to these databases to get user credentials including second factor authentication (2FA) credentials to access these systems
b. Advanced Persistent Threats (APTs) that seek to install malware such as RATs for exfiltration of this data to a remote server under the Command & Control (C&C) of the attacker. APT's will often target encryption keys or supplementary data that would aid in decrypting the stolen database in order to obtain the plaintext cardholder data.

Payment fraud

Payment fraud detection should occur at different layers and systems involved in processing mobile payment transactions. Issuers are responsible to enforce controls to prevent use of stolen credit card data to be used by the mobile payment users to:

a. conduct fraudulent mobile payments transactions
b. enforce credit card limits on the payment transactions themselves
c. on the debit cards amounts linked to consumer direct bank accounts managed by the issuer bank[17]

Token data compromise

Since issuers can choose to leverage the tokenization service from the payment networks or implement their own token service and become a Token Service Provider themselves, they will be at increased risk of threats against token data confidentiality, integrity and availability.

[17] **Note**: at the time of issuance of this paper liability for mobile payment fraud is taken by the mobile payment organizations (e.g. Apple for Apple Pay)

3.10 Mobile Payment Applications Providers (Servers & Cloud Services) Threats

Possible threats and attacks against services provided by mobile payment application providers (e.g. Apple, Google) include:

Compromise of cardholder's sensitive data

Attackers might direct their effort to cardholder credit/debit data and personal data of the user that is stored by the mobile payment service provider. The main motivation behind these attacks is to steal credit card data as has been discussed previously.

This data compromise might also occur during transmission of cardholder sensitive data from the mobile device to the servers such as during registration of the mobile payment application service with the card issuer[18].

Compromise of the user profile managed by the mobile payment service provider

Since the mobile application has access to the mobile payment servers such as during card enrolment, an attacker could seek to compromise this access to commit fraud such as:

a. to enrol stolen credit data with the mobile card enrolment service
b. to abuse non authorized access to the user profile managed at the mobile payment provider (e.g. through stolen/lost device or through online access to his/her account)
c. to change account profile contact details, emails, phone numbers etc. to facilitate fraud

Token service data compromise

Since mobile payment providers can also implement their own token service they are also at risk of threats against:

a. the token management process that encrypt and decrypt tokens
b. the management of keys
c. the integrity and availability (e.g. denial of service) of the tokens issued for payment authorizations

DDoS attacks

Digital wallet services including cloud services used by mobile payment providers can be targeted with DDoS attacks by threat actors seeking to disrupt mobile payment services. These DDoS attacks might affect transactions that require real time access by the mobile payment application to the payment services hosted in the cloud such as for the initial mobile payment card enrolments.

[18] **Note**: in the case of Apple Pay and Google Pay with Secure Element mode, credit cardholder and personal data is not stored on Apple or Google Servers but on the Secure Element of the mobile device. In the case of Google Pay used in HCE (Host Card Emulation) cardholder and personal data is stored in cloud servers and might be subject to compromise of cloud services with malware and exploitation of possible vulnerabilities in the software implementation of digital wallet services.

3.11 Potential Vulnerabilities of Digital Wallets

Enrolment

The first step to use a mobile payment is the enrolment of the user's credit cards into the app. The provider cannot, of course, know whether the card entered belongs to the user or not. This is something that only the card issuer can know. Providers facilitate issuer's decision making by providing information. For example, Apple Pay passes on information to the card issuer including the user phone number, location (if Location Services are enabled), iTunes account information, etc. The issuer has to make a decision (based on an automated risk assessment) of whether the card is accepted or not.

A recent vulnerability[19] exploited by fraud rings abuses a weakness in the way the risk assessment is performed and some issuers have accepted stolen cards to be added to Apple Pay accounts. Once accepted, they were used to buy products which were in turn sold online. By Apple's rules, it's up to credit card-issuing banks to verify the legitimacy of their cards when they're added to Apple Pay, a process called "provisioning". Apple Pay relies on fraud detection by the card issuer for fraudulent card enrolments with the digital wallet. Fraudulent transactions have been possible because of weaknesses in fraud traceability to track fraudulent card registrations back to the user that either registered the card. In the case of payment fraud, liability for fraud might shift from the banks to merchants is based upon the network payment policies and the methods used to verify the user[20].

Credit card entry

Another potential attack vector is when the credit card information is initially entered into Apple Passbook/Google Wallet / Samsung TEE, which uses the phone's camera. If the phone is already infected with memory malware or other malware that has compromised the camera or passbook/wallet the information can be stolen using memory scraping, OCR recognition or even by sending the raw image capture for offsite analysis to C&C servers.

Additionally, the data may be eavesdropped in transit when the credit card information is sent from the device to servers in the cloud. Should the device be compromised, an attacker may be able to gain access to the network traffic and therefore the credit card information.

An attacker could exploit a social engineering attack by requesting a user to re-enter the credit card details. A recent bug in iOS allows an attacker to replace a legitimate application with a clone developed by the attacker[21], enabling a MITM attack. The attacker could masquerade passbook and steal card information.

User authentication

Providers rely on fingerprint biometrics for user authentication. Extensive research has proven that fingerprint authentication can be bypassed and has been shown to be breakable[22]. If the user's phone is

[19] Drop Labs, "Rampant: Explaining the Current State of Apple Pay Fraud", http://www.droplabs.co/?p=1231
[20] Apple, "About EMV and Apple Pay for Merchants", https://support.apple.com/en-us/HT205645
[21] Security Tracker, "Apple iOS Multiple Bugs Let Remote Users Execute Arbitrary Code", http://www.securitytracker.com/id/1029888
[22] Kaur, Manvjeet, Sanjeev Sofat, and Deepak Saraswat. "Template and database security in Biometrics systems: A challenging task." International Journal of Computer Applications 4.5 (2010): 1-5.

stolen, it would not be hard to bypass the biometric authentication, which unlocks the entire phone and financial payment process.

For example, in Google pay users can authorise payments just by entering the lock screen pattern. As this is a mechanism which can be easily eavesdropped, it may encourage opportunistic attackers to steal a device in order to perform payments on behalf of the victim.

Fraudulent payment transactions

Card issuers may not accept liability for fraud if their terms and conditions are not met. For example, Lloyds Terms and Conditions[23] state "ensure you only register your own fingerprints and not anyone else's" which would potentially invalidate a fraud claim if more fingerprints are registered on the device.

Accountability for payment transactions

Payment providers require fingerprint authentication to perform the payment. However, a number of individuals may have been enrolled in the fingerprint database and therefore anyone who can authenticate to the phone would be able to perform a payment. If a number of users have access to the device, it creates an accountability failure as it is not possible to uniquely identify the person who performed the payment.

Third party trust

Regardless of the mobile payment provider, enrolling on the system requires a certain level of trust on the third party. Although, it is likely that the third party is doing due diligence and the required security mechanisms are in place, the consumer does not have full assurance that the data may not be compromised at some point in the future.

Apple offers the functionality to integrate with Apple Wallet, so that third party applications can perform payments, organize vouchers, etc. This is done by an API called PassKit. A malicious application could be installed on the device which would in turn access PassKit to manage credit cards or perform unwanted in-app payments or misuse passes.

Wider attacking surface on a stolen device

Should a mobile is stolen, attackers may be able to gain further access to payment cards, which would introduce a new incentive to steal the device.

Phishing attacks

The wider adoption of mobile payment technology may encourage attackers to perform attacks impersonating legitimate applications and requesting credit card data to the user, in an attempt to lure a user to disclose the information.

Tokenization Services

[23] LLoyds Bank, "Using your LLoyds PLC Card with Apple Pay - Important Information", https://www.lloydsbank.com/legal/payment-service-4/terms.asp

Tokenization services will become a single point of failure, something similar to DNS infrastructure. For that reason, DNS has been designed with redundancy in mind, whereas tokenization services may not. Additionally, they will become a prime target as they will map real PANs.

Tokens eavesdropped in transit which resulted in a successful transaction cannot be reused and therefore cannot be used to perform another payment. Eavesdropped valid tokens which have not been used in a transaction could be reused.

For example, in Samsung Pay, analysis of the tokens show that they are incremental and future tokens are vulnerable to a specific attack[24], which could potentially allow an attacker to automatically brute force valid tokens by extrapolating future tokens based on observed tokens. Furthermore, sniffed tokens can be reused during a limited window of time of 60 seconds. This would allow attackers to perform a valid payment over the Internet, away from the victim's device.

Mobile as a target

Mobile devices normally are not subjected to the same level of protection as desktops. For example, they rarely run antivirus, firewall, etc. Introducing new services for payment will make them a more interesting target to attackers.

Researcher Joshua Rubin with Zvelo[25] successfully performed brute force attacks on the PIN number (4 digits) of Google Wallet. Rubin also was able to retrieve the Google Wallet PIN number from rooted devices.

Implementation Issues

In a competitive market, all the payment providers are not going to stand still. It is envisaged that new functionality will be continuously released. As such, there is a risk to run potentially immature code which may be prone to security issues.

For example, Apple Watch lets users access to Apple Pay without any further authentication if the watch is not removed from the wrist. However, there is an example[26] that show that it is possible to remove Apple Watch from the victim's wrist without locking it. This attack could be performed to gain unauthorised access to the user's Apple Pay.

In another example using Google Wallet, ViaForensics[27] was able to root the phone to access the Cardholder name, last 4 digits of the credit card, and expiration date even after Google Wallet was reset. Prepaid cards for Google Wallet were exposed when Google Wallet was wiped and when setup again to a new account all prepaid cards were accessible. This flaw has since then been fixed by Google. Following

[24] Salvador Mendoza, "Samsung Pay: Tokenized Numbers, Flaws and Issues", https://www.blackhat.com/docs/us-16/materials/us-16-Mendoza-Samsung-Pay-Tokenized-Numbers-Flaws-And-Issues-wp.pdf

[25] Zvelo, "Google Wallet Security: PIN Exposure Vulnerability", https://zvelo.com/google-wallet-security-pin-exposure-vulnerability/

[26] Wonder How To, " Apple Watch Vulnerability Lets Thieves Use Apple Pay Without Your PIN", http://ios.wonderhowto.com/how-to/apple-watch-vulnerability-lets-thieves-use-apple-pay-without-your-pin-0161940/

[27] http://viaforensics.com/mobile-security/forensics-security-analysis-google-wallet.html

this, Google announced that the Google Wallet is not supported on rooted devices and took steps to protect cardholder sensitive data[28].

Samsung Pay also had problems with their payments using traditional magnetic stripe POS terminals. In order to emulate the payment, the Samsung smartphone creates a magnetic signal which is received by the card reader. This communication is performed in clear text and therefore a suitably placed attacker could gain access to the information sent to the POS. A recent research[29] claims that the information transmitted from the phone to the POS when using MST is stronger than required, allowing an attacker to successfully eavesdrop the token with a receiver placed at more than 2.0 meters away from the victim's device.

[28] MIT Technology Review, "Is Google Wallet Safe ?", https://www.technologyreview.com/s/426921/is-google-wallet-safe/

[29] Daeseon Choi, Kongju National University of Korea and Younho Lee, SeoulTech Korea, "Eavesdropping one-time tokens over magnetic secure transmission in Samsung Pay", https://www.usenix.org/system/files/conference/woot16/woot16-paper-choi.pdf

4. Recommendations

The following list covers the key recommendations that should be followed by vendors providing mobile payment applications:

4.1 Minimum Security Measures

Mobile payment providers should make customers and merchants aware of the risks and consequences of running their application in a mobile environment.

Customers should at least follow a number of minimum security measures that should be required to securely use their application:

- The customer should update the Operating System on a regular basis, as soon as the OS provider makes an update available.
- Network transport should be trusted: Performing mobile payment transactions from an untrusted network (such as public WIFI hotspot) could facilitate third parties intercepting the communication and potentially tampering with the payment.
- Customer authentication to the mobile device should always be enforced with the use of biometric controls or strong PIN/pattern.
- Effective configuration should be in place in case the device is lost or compromised, such as remote data wipe out.

Merchants should also follow guidelines to ensure the security of the payment transactions:

- POS software should be updated as soon as the provider releases a security update. POS software has visibility of all payment transactions, and therefore sits on a privileged place for an attacker. POS have been targeted by malicious software and therefore providers should update the terminals to ensure that the software preserves its integrity.
- POS could be tampered with from a hardware perspective. Merchants should be made aware of potential attacks so that they can be efficiently remediated

It is also required that the mobile payment application is built with security in mind, where appropriate **secure software development** should be adopted. As a minimum[30]:

- Avoiding hard-coded sensitive information such as passwords or keys.
- Employing anti reversing techniques.
- When possible, verifying the integrity of the running code, to ensure that it has not been back-doored. This includes the importance of provisioning the application only through trusted application stores.
- When possible minimizing at all times potential man-in-the-middle attacks by implementing effective certificate pinning to ensure that the application is communicating to the intended end points.

[30] Smartphone Development Guidelines, ENISA 2016, https://www.enisa.europa.eu/publications/smartphone-secure-development-guidelines

4.2 Security of the Payment Chain in the Ecosystem

Every party involved in the mobile payment ecosystem should be able to show evidence of following due diligence with regards to security. This includes not only the mobile payment provider, but also key actors such as the TSP (token service providers) and cloud databases.

It is important that the end-to-end security review is not performed in isolation, where each element is reviewed individually, but also the integration between parties. It is therefore necessary that the different parties collaborate not only in the integration and delivery of a service, but also in ensuring a common goal towards increasing security. It is also key to review the security from all the threat actors, such as privileged access provided to internal employees or system administrators.

This holistic security review should be based on a threat model of the ecosystem where as a minimum the areas covered in Annex B should be covered. It includes specific component level threats, vulnerability and security measures/controls.

Evidence of this collaborative security programme should be made available to customers, so that they can make informed decisions as to the level of commitment to security of each provider.

4.3 Mobile OS security

Mobile OS providers should ensure that their OS is regularly updated to fix any security issue identified, which may jeopardise the integrity, confidentiality or availability of the system or data. They should also deploy a robust model and implement reliable mechanisms to securely store confidential information.

Additionally, mobile OS providers should provide effective means to prevent the use of jailbroken devices for mobile payment applications, as a jailbroken device could potentially break the security model enforced by OS security controls.

Mobile OS providers should allow for possibility to undisputedly identify which fingerprint was used for authentication and/or authorization.

4.4 Transparency of Security Measures

Mobile payment application developers should provide visibility to the security measures applied to the application when offering it to the clients.

Providing visibility to consumers of which safeguards are taken by the all mobile payment application stakeholders (i.e. mobile payment providers, merchants, payment processing services, acquirers, issuers and card organizations) to protect consumers confidential, personal and payment data will ultimately lead to a larger amount of transactions occurring over the mobile channel and thus increase adoption.

4.5 Effective Risk Management Program

An effective risk management program should be in place that focuses on mitigation of mobile payment application risks and identify measures including detection of possible data compromise and fraud. To this end, mobile payment providers should have a reliable and accurate fraud monitoring system which reliably detects transactions outside the customer’s baseline, due to for example a stolen mobile device being used by an attacker. They should also be able to effectively prevent further payments from a compromised mobile payment account.

Risks shall be reviewed at every change being introduced in the mobile application to identify control weaknesses/gaps and vulnerabilities. These risk reviews shall be ongoing, considering the emerging and evolving threats targeting the mobile payment application ecosystem.

As a minimum it is recommended that the security measures included in this paper shall be assessed as basis of high level reviews based upon the high level risks listed in Annex A and threats and countermeasures in Annex B as the scope for a mobile security risk management program listed in Annex C.

Annex A: Mobile Payment Application Most Common Security Risks

The Open Web Application Security Project (OWASP) provides a list of top 10 mobile security risks[31] . These risks can also be considered as mobile payment application security risks with examples included herein:

OWASP T10 Mobile Security Risks	Mobile Payment Application Security Risks
1- Weak Server Side Controls	Mobile payment applications rely on the mobile application vendor servers hosted usually on the cloud to perform critical security operations. Apple Pay servers for example provide functionality such as re-encrypting payment credentials for payments within apps, controlling the state of credit and debit cards in Wallet and the Device Account Numbers stored in the Secure Element. In Google Wallet used in HCE mode, sensitive data is stored in databases hosted in a secure cloud environment. The security of these critical servers depends on several controls that are the responsibility of the vendor to enforce. When these controls (e.g. fraud detection) cannot be enforced by the mobile payment application vendor it is important that liability clauses covering cases such as fraud and data breaches are clearly spelled out in Service Level Agreements (SLA) between the vendor and any contracted 3rd parties involved. This includes merchants, merchants banks, card issuers and card issuers banks and the various payment card network operators.
2-Insecure Storage	Securing confidential data such as cardholder personal identifiable information as well as sensitive information such as the PAN, CVV, tokens and cryptograms is one of the most critical mobile application security controls. By design some mobile payment applications use the secure element to store confidential cardholder data and to process sensitive data to authorize payment transactions. Use of tokens as a replacement of PANs as well as of virtual credit card data instead of the real credit card data is another way to secure the confidentiality of the real data. The security of the token, its generation, transmission and verification are of paramount importance. Ensuring that the token is non deterministically random and that the token can only be used once, helps safeguard against multiple types of attacks that rely on token replay.

[31] OWASP, "Projects/OWASP Mobile Security Project - Top Ten Mobile Risks", https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks

	When the tokens themselves are stored in a cloud environment as is the case with Google Wallet and HCE, an additional dimension of risk is introduced that must be assessed using traditional cloud security assessment frameworks. e.g., is it a multi-tenant system or dedicated cloud infrastructure, what are the interfaces, what are the controls protecting data in motion and at rest to name a few.
3-Insufficient Transport Layer Protection	Insufficient protection of data in transit starts from the transport security of the channels (e.g. NFC, MST, BLE) used for contactless mobile payments as well as client to server channels (e.g. SSL) used for provisioning the mobile application and transporting cardholder data. The main challenge with mobile payments as well as with credit/debit card payments using POS terminals is that the security of the data in transit for authorization and for processing of payments falls under the control of different parties which includes the mobile payment app vendors, the merchant, the card issuer and last but not least the card payment network. End to End Encryption (E2EE) is an excellent control which compensates for the risk of insufficient transport layer protection because of misconfigurations or vulnerabilities in the transport layer at any of the tiers of the payment processing network architecture. Utilisation of technologies such as SSL certificate pinning helps further safeguard against Man-in-the-Middle attacks.
4-Client Side Injection	The mobile payment UI itself represents another vector of attack for exploitation of input validation vulnerabilities by attackers. Mobile application input validation vulnerabilities that allow client side injection of data or executable code (e.g. JavaScript) need to be validated during SDLC assurance activities and remediated prior to production roll-out by the mobile payment vendor be it, a financial institution or a hardware manufacturer such as Apple, Samsung.
5-Poor Authorization & Authentication	Authentication and authorization are critical controls for every mobile payment application since they not only authenticate the user but also authorize the payment. Weaknesses in authorization might allow impersonation attacks with stolen data such as stolen tokens used by a different device and user that they were intended for. Here, transaction verification plays a vital role and needs to complement tokenisation efforts. The token should not only be one use only, but also tied to the device with other heuristic attributes such as GPS location baked in. GPS when tied to the transaction would also help indicate whether the token is being sent from a geolocation that the user is not traditionally associated with which may indicate fraud (or travel) but more importantly help detect 'superman' attacks whereby the legitimate user pays for a good or service in e.g., London and 5 minutes later, the same token is seen to be sent from Brazil, more likely forming a good indicator of fraud.

	Attackers attempt to bypass authentication by spoofing biometric data, attempting to reset user credentials when they are lacking strong verification of the user with additional validation before reissuance of PINs and passwords. An additional common attacker attempt includes decompiling the application while searching for hard-coded passwords and PINs. Weaknesses protecting authentication data in storage e.g., in the SQLite databases within the application directories, is also a commonly observed design flaw that can be exploited by attackers to often compromise authentication should the attacker have physical possession of the device or have gained persistent root level access to it remotely and authentication data is stored within with little or no protection.
6-Improper Session Handling	Improper session handling can be caused by failing to invalidate the session at logout, poor implementations of session expiration, issues with session tokens/cookies such as replay and hijacking of the sessions because of lack of protection of session data such as cookies in transit between client and servers. Similar types of improper session handling might affect a web based mobile payment application. For mobile payment applications that use tokens for authorization of payments some of the security requirements that are applied to web session tokens can also be applied such as randomness, freshness to prevent replay, limited validity and expiration time being set in addition to the usual controls about ensuring that the tokens themselves are sent over encrypted transport.
7-Security Decisions Based Upon Untrusted Inputs	Every time a user is prompted to enter data, input should be treated as untrusted and should not be blindly accepted by device or backend API's and blindly acted upon. Each client input should map into a regular expression of accepted boundary conditions and conform to that input. This should then be verified on the server side to ensure that the data entered in the UI and transported to the API endpoint still conforms to the regex boundary conditions as an attacker may interact with the API directly bypassing any business logic checks on the frontend mobile client.
8-Side Channel Data Leakage	Attackers could access sensitive data using side channels such as by installing malware on the device in order to control it remotely or to steal data from the device. Preventive measures include application isolation such as sandboxing and virtualization monitoring of mobile application on the device. Detection measures include Jailbreak detection, malware detection and secure provisioning of the mobile application and third party libraries used by the application.
9-Broken Cryptography	This risk includes a wide range of categories that might include non-secure key storage such as hard-coding of keys in the mobile application source code

	and/or configuration files as well as insecure use of crypto such as lack of using secure random seeds, use of weak encryption algorithms and key lengths, insecure key generation and entropy. A secure design review and secure code review including automated static source code analysis can identify some of these issues (e.g. hard-coded keys and use of insecure algorithms). Mobile devices have traditionally suffered from low entropy generation which may generate bias towards specific low entropy crypto keys due to the constricted resources both processing, memory and battery available however this is gradually getting better when physical sensors are tapped into.
10-Sensitive Information Disclosure	Unauthorized access to sensitive data that is stored by the mobile application might occur in the case of a device being stolen, lost or compromised (e.g. with malware/rootkit). In such cases it is appropriate to assume that even encrypted data on the device using device encryption (e.g. keychain) can be compromised. Lack of encryption of sensitive data is also a security design flaw that can be detected early through threat modelling and source code review of the mobile payment application. Sensitive information disclosure might occur because of caching and logging of confidential data of the user as well as tokens by the application as well as third party libraries used by the mobile application. A likely attack vector for sensitive information disclosure is attacking the user of the mobile application with social engineering.

Annex B: Threats, Vulnerabilities and Security Measures/Controls

We have summarized herein the possible threats previously analysed with the threat model by following a risk centric threat modelling methodology[32] as well as the identified mobile payment applications vulnerabilities (e.g. vendor specific and common for mobile applications) and mapped to possible measures/controls that can be applied to the various components of the mobile payment application ecosystem to mitigate the mobile payment application risks.

Mobile Payment Component	Possible Threats	Possible Vulnerabilities, Design Flaws & Security Mis-configurations	Possible Security Measures/Controls
Users/Cardholders	Phishing and social engineering	Lack of user's due diligence validating content in emails, messages, SMS being trustworthy before selecting URLs, downloading attachments	Security awareness, education and communication
	Inadvertent installation of rogue applications packaged with malware/rootkits	Use of mobile payments with public Wi-Fi connections	Do not use public Wi-Fi hotspots for mobile payments
		Missing following minimum security hygiene rules, using jailbroken OS (e.g. to install untrusted applications and files on device)	Keep OS up to date Do not jailbreak phone
Mobile Devices	Unauthorized access to lost or stolen mobile devices	No PIN lock set PINs set to a weak PINs No remote device lock set No remote data wipe set	Remote device lock Remote data wipe PIN lock Strong PINs User to device biometrics authentication factors (e.g. fingerprint, iris)
	Data interception via installation of spyware	Not up-to-date OS Jailbroken device Zero-day vulnerabilities	Keep OS up to date Keep default security controls & measures on device

[32] Marco M. Morana and Tony Ucedavelez," Risk Centric Threat Modelling: Process for Attack Simulation and Threat Analysis", http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470500964.html

Mobile Payments & Digital Wallet Applications	Reverse engineering the application source code	Hardcoded secrets (e.g. private keys) Missing to disable code debugging routines	Adopt secure coding practices and secure code reviews (manual and automated via tools) Source code obfuscation Jailbreak detection Anti-debug protections
	Tampering with the mobile payment application source code, repackaging rogue application executables	Unsigned production binaries	Integrity source code protections White-box cryptography Secure application provisioning through trusted application stores Takedown rogue applications from unauthorized application stores
	Exploit of mobile payment application vulnerabilities and design flaws	List of possible exploits against mobile payment applications (Apple Pay, Google Wallet and Samsung Pay) refer to the section of this paper: Vulnerabilities/Design Flaws of Digital Wallets) summarized herein: 1) Credit card provisioning weaknesses (adding stolen credit cards, use of raw images to enter sensitive data) 2)Weaknesses in biometric identification for initial authorization of transactions (e.g. fingerprints not tied to user payment transactions but user to device authentication) 3) S/W vulnerabilities and weaknesses in third party applications (including APIs) that provide access to digital wallets	Identity, validate/test and remediate application design flaws and vulnerabilities and Top 10 Mobile Application Security Risks in Addendum A

		4) Weaknesses in payment authorization provisioning with mobile paired smartwatch device 5) Credit/debit card not stored encrypted in SE or processed in TEE 6) Weak PINs exposing them to brute force attacks 7) Insecure communication channels with POS contactless terminals 8) Insecure tokens used in MST connections 9) Inadequate signal strength for MST processing	
Merchants (e.g. stores)	Uploading malware on the POS contactless payment terminals and POS servers	Use of default password to access POS terminals (available online) POS and POI security mis-configurations and security hygiene (e.g. keeping software up to date, patching systems)	Change default passwords on POS systems and keep POS software up to date
	MiTM against POS contactless terminal and POS point to point connections	Insecure connections between POI and POS	Use SSL between POS connection point (POI to POS)
	Relay attacks against NFC enabled POS contactless terminal	Insecure access to LAN and to POS systems Lack of enforcement of minimum privileges for POI and POS access	Deploy and configure firewalls Restrict POI and POS access to authorized users

Payment Service Providers	Compromise of S/W running on contactless terminals Compromise of S/W installed on POS Servers Compromise of Payment Gateways	Design flaws and un-patched S/W vulnerabilities in POI terminal/credit card machines and POS systems and payment gateways to/from acquirers	Secure by-default design, vulnerability testing, patching of POI terminal (card machines) H/W and S/W. Fix S/W vulnerabilities in POI, POI and payment gateways hosted at the payment service providers
	Data connectivity (merchant hosted POS connection to PSP and from PSP to acquirer)	Insecure point to point connections between merchant POS server and PSP and between PSP and acquirers	Enforce secure point to point connections (between merchant POS and PSP and between PSP and acquirers)
Acquirers	Payment processing systems compromise	Un-authorized access to payment processing systems/applications and weaknesses in enforcement of internal security controls and measures to access these systems	Enforce high security standard measures for payment processing systems and 2FA for user authentication/access Enforce minimum privileges for user access
	Installation of malware/RAT for APTs	Non effective malware detection, data leakage detection/prevention and fraud detection/prevention	Deploy malware detection, data leakage and fraud prevention
	Data connectivity (external from acquirer to issuer and internal among servers) compromise	Insecure external and internal point to point system connections Weak server to server authentication among internal systems	Secure internal point to point connections with SSL/mutual authentication
	Repudiation of mobile payment authorizations	Gaps in non-repudiation controls for processing authorizations such as out of band verification/confirmation of suspicious transactions and digital signing of transactions	Require digital signatures to sign and verify payment authorizations from issuer

Payment Network Providers	Token services provider services & servers compromise	Misconfiguration of servers providing tokenization services Non secure key storage (e.g. use of non-encrypted file storage instead of HSM) Insecure user access to the token vault (where token to PAN mapping is provided in lookup tables)	Secure configuration and hardening of critical servers Secure key storage in hardware encrypted security modules (HSM) Dual controls and strong authentication 2FA to access the token vault.
	Data connectivity compromise	Insecure connections to/from acquirers and issuers	Enforcement of E2EE (End to End) encryption for protecting cardholder data in transit to issuer.
	Denial of payment settlement services	Weaknesses in protection of Denial of Service (DOS) attacks against TSP service	Anti-DOS measures (application and network layer) to protect token services
Issuers	Payment authorization process compromise	Weaknesses in enforcing strong authentication for access to critical systems and databases where cardholder data is stored for validation and payment authorization to acquirer	Enforce strong multi-factor authentication for access to critical systems where credit cardholder data is being stored. Enforce minimum privileges for users that have access to internal critical systems used for verify cardholder data and authorize payments based upon specific business rules
	Confidential cardholder data compromise through malware/APT Payment fraud Token services compromise (if TSP is hosted optionally by the issuer, refer to token services	Non-effective malware detection and prevention measures Misconfiguration of fraud detection systems including rules such as positive payment checks, max limit amount per transaction, daily limits, velocity tagging	Deploy malware detection and prevention, suspicious activity detection rules based upon aggregated log analysis Configure fraud detection and prevention systems and enforce fraud management rules for mobile payment transactions

	provider services & servers compromise for payment network providers for vulnerabilities and measures)		
Mobile Payment Applications Providers (Servers & Cloud Services)	Compromise of cardholder's sensitive data	Weaknesses and vulnerabilities on digital wallet servers and applications hosted at the mobile payment application provider	Enforce information security policies and processes requiring identification and remediation of vulnerabilities in servers and applications
	Compromise of user profiles managed by the mobile payment service provider	Absence of malware detection and prevention on critical servers that provide access servers where cardholder data and user profiles are stored. Gaps in deployment of 2FA to access servers and maker/checker controls	Deploy malware detection and prevention measures Enforce 2FA for internal user's access to critical servers such as digital wallet services where cardholder data and user profile information is stored. Enforce user entitlements and minimum privileges
	Enrolment of stolen credit card data for use of mobile payment by fraudsters	Absence of fraud detection and prevention for use of stolen credit card holder for enrolment in mobile payment applications	Deploy fraud detection and prevention for high risk functions such as change of account profile, credit card enrolment and payment transactions
	Denial of Service (DoS) attacks	Weaknesses in anti-DoS measures to prevent DoS against digital wallet and account profile services hosted in data centers and cloud services	Deploy anti-DoS measures for critical servers hosted in data centers and in the cloud

Annex C: Risk management

C.1 Strategic Risk Management

Enhancing security of mobile payment applications ultimately enhances consumer trust and this in turn will act as a catalyst for growth of mobile payments. This trust can be “earned” by providing visibility of the various security measures and controls that have been deployed to safeguard cardholders and customer’s data privacy. This visibility also helps mobile payment developers and mobile payment providers to engineer measures that can reduce the likelihood and impact of cyber threats exploiting vulnerabilities, weaknesses and gaps in security controls of mobile payment applications.

Today mobile payment application developers are on the forefront of the task to develop mobile payment application software that is secure by design and by implementation. During design it is important to follow security by design principles . Specifically, for the design of secure mobile payment applications it is important to avoid design flaws that could impact the security of the mobile payment application and increase the risks of an attacker exploiting them to gain access to confidential cardholder data, confidential PII data and financial data.

In Addendum A of this research paper, we provide a list of most common mobile payment application risks. This list constitutes initial guidance for deriving non-functional security requirements that can be followed by mobile payment application providers during design and implementation as well as identification and remediation of most common mobile payment application vulnerabilities.

Besides following an application security programme focused on security by design and testing of mobile payment applications it is important to identify and apply countermeasures that mitigate the risks of specific attack vectors targeting mobile payment applications and the various assets of the mobile payment ecosystem. A threat, vulnerability control framework of the mobile payment application ecosystem might constitute the basis to analyse the various types of risks affecting the ecosystem and make recommendations for mitigating these risks when vulnerabilities and controls gaps are assessed (e.g. by a high level risk and control security assessment). Such framework is provided in Addendum B

One of the aims of this paper is to provide risk management recommendations for the different parties that own the various components and assets of the mobile payment application ecosystem. Before recommendations are made, it is important to highlight the various risk management strategies that can be followed for making risk management decisions such as for risks avoidance, risk acceptance and risk mitigation.

A risk management decision might consist of not storing sensitive data on the mobile device: the risk strategy rationale for this decision might be reducing the opportunity for an attacker to access sensitive data stored on the mobile device. If a decision is made to allow storage of sensitive cardholder data and payment data on servers hosted in the cloud, the main question is to whether cloud based security measures and controls (that are mostly software controls in cloud SaaS) are strong enough to safeguard aggregated confidential cardholder data of several mobile payment users stored on cloud servers.

Another decision could be to allow storage of sensitive data including secrets such as encryption keys on the mobile device instead, relying on the level of assurance of the security provided by both hardware and software security controls on the device rather than having an aggregate central point of failure on the cloud.

An important risk management strategic decision in this case is to decide to whether leverage hardware security on the mobile device when software security controls cannot be trusted to be secure enough in case of specific attacks such as in case of malware compromise.

Deciding not to store aggregated fingerprint biometric data of customers on a secure server/vault is a strategic risk decision that reduces the risk of targeted attacks against aggregated authentication and user identification data (e.g. fingerprints). The same risk applies to storing private keys or millions of authentication data such as passwords and PINs on a database encrypted as salted hashes, as opposed to storing each one in a secure element that never leaves the device whose authentication and identification is under the controlled secure code execution of the hardware based security of the device TEE or Secure Enclave.

Another important risk strategic decision is on the value of data as an asset that needs to be protected and to think in advance of what damage an attacker could do (e.g. that is what the data can be abused and how would that data be monetized) in the event this data/asset is being stolen or compromised. If the data that needs to be protected is cardholder data used to authorize payments such as for example the Card Validation Values (CVVs), card expiration data and the PAN, replacing this data with virtual credit data (e.g. an alias PAN) and tokens, that could not be used to counterfeit cards or for card non present (e.g. on-line purchases) transactions, solves several problems in relation to keeping this data encrypted in storage and transit through the different components of the mobile payment ecosystem (e.g. mobile device, POS contactless terminals, payment processing services, acquirers and the issuers). This is the strategy that has been followed by the mobile payment providers we have discussed in this paper.

When the strategic decision has been made to store confidential and sensitive data of customers, it is important to make a risk decision on whether the security controls applied to protect confidentiality, integrity and availability of sensitive data including secrets such as encryption private keys, are strong enough to mitigate the risk of threat actors seeking to compromise this data. Typically threat actors invest time, energy and resources in attacks including investments in cybercrime tools that maximize their reward and minimize their effort.

In the case of mobile payment applications, it is unlikely that a threat actor would attack one mobile device at a time just to steal a token and cryptogram that could be used only once. The attacker would rather try to exploit security weaknesses in contactless POS terminals and POS servers to compromise token and cryptograms in transit or attack the Token Service Providers where tokens and secret keys are stored.

To limit the possible fraud of mobile payment initiated transactions executed over contactless terminals a sound strategy is to limit the maximum value of each transaction to small amounts (e.g. 30 GBP in UK) when user to device identification such as fingerprint (e.g. Touch-ID in Apple Pay) and iris biometric identification (e.g. in Samsung Pay) are not used (e.g. in devices that only allow user to device authentication with PINs).

Mobile payment application providers should also make sure they apply strong authentication and identification for protecting abuse of high risk account management functionalities such as for contact profile changes, changes of the credit/debit card data linked to the account and changes of personal details and contact information such as address, emails and phone numbers used for billing and for payments confirmations and notifications.

When designing mobile payment applications that leverage user to device authentication factors that the mobile device has implemented, it is important to make use of industry standards for Unified Authentication Factors (UAF) and for Unified Two Factor authentication (U2F) such as the ones promoted by the FIDO Alliance[33]. One important reason to leverage these standards besides interoperability and information assurance is consumer's privacy since these standards help to implement authentication by taking into consideration the need of user privacy [18]

An additional strategic risk decision to make in advance of any deployment of mobile payment application is to consider the possibility of data compromise and in the eventuality that this compromise is being detected which actions can be taken to prevent further impacts. For example, if anomalies are detected as suspicious user behaviour such as attempts to change the user profile, contact information, unusual volume of payment transactions, spending from different geographical locations and others can be set in fraud detection rules and trigger actions in response such as issuer bank calling to validate the transaction to check that is originating from the intended user/cardholder.

In general, in order to prevent further impact, it is necessary to detect different events, correlate them and analyse them. Mobile payment providers that host most of functionality in the cloud have the opportunity to leverage large transactional datasets of user aggregated data and apply machine learning and artificial intelligence to identify possible data compromises and fraudulent transactions.

C.2 Risk Mitigation Strategies

In the risk mitigation strategy, it is important to consider the possible impact to the business derived by the loss of consumer's personal data and privacy. Making consumers aware of the level of security being provided to safeguard their privacy should be one of the main priorities for mobile payment application providers as well as for the other entities of the mobile payment ecosystem.

In the effort to raise awareness on mobile security risks, ENISA has published documents on mobile identity management and recommended mobile users controls over privacy settings including privacy preservation on mobile payment information such as shopping list and history[34].

In terms of guidance on the security of mobile payments in Europe, a document issued by the European Central Bank (ECB) Forum on the Security of Retail Payments[35] also recommends that mobile service payment providers provide security awareness, education and communication to mobile payment users and to follow a risk mitigation strategy focusing on the identification and assessment of risks on an ongoing basis and to focus on protecting and securing sensitive payment data.

Compliance risks should also be a priority as the risk of unlawful non-compliance with privacy regulations such as the General Data Protection Regulation (GDPR) in EU (Regulation EU 2016/679[36]) might have

[33] FIDO Alliance, "Specifications Overview", http://fidoalliance.org/specifications/overview/

[34] ENISA, "Mobile Identity Management", https://www.enisa.europa.eu/publications/Mobile%20IDM

[35] European Central Bank, "Recommendations For The Security of Mobile Payments", https://www.ecb.europa.eu/paym/cons/pdf/131120/recommendationsforthesecurityofmobilepaymentsdraftpc201311en.pdf

[36] European Union, "Proposal for a Regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)", http://data.consilium.europa.eu/doc/document/ST-9565-2015-INIT/en/pdf

negative tangible (e.g. fines) and intangible (e.g. reputational damage) impacts on mobile payment application providers and they should therefore be treated as high priority risks.

Consumer privacy requirements such as the requirement to implement consumer privacy controls by design to safeguard consumers Personal Identifiable Information (PII), contact information and credit/debit card information including shopping habits and location data, should be in scope for mobile payment applications and digital wallets that collect, store, and transmit sensitive data.

Evidence that EU citizen’s privacy requirements are satisfied as part of the security by design and implementation of the mobile payment application should be also asserted by industry standard and/or governmental vetting bodies to help ensure that an app conforms to such mobile payment security and privacy requirements prior to it being made widely available for public use.

3rd party and/or governmental e.g., European/ENISA level vetting of mobile payment applications should be a requirement for both mobile payment companies operating in the EU and also mobile payment companies who in the course of their business, process transactions for European citizens.

ENISA

European Union Agency for Network
and Information Security
Science and Technology Park of Crete (ITE)
Vassilika Vouton, 700 13, Heraklion, Greece

Athens Office

1 Vasilissis Sofias
Marousi 151 24, Attiki, Greece

TP-05-16-075-EN-N

PO Box 1309, 710 01 Heraklion, Greece
Tel: +30 28 14 40 9710
info@enisa.europa.eu
www.enisa.europa.eu

ISBN: 978-92-9204-199-1
DOI: 10.2824/67533