Towards trusted mobile payment services: a security analysis on Apple Pay

Ashay S. Jawale* and Joon S. Park

School of Information Studies,
Syracuse University,
334 Hinds Hall, Syracuse,
New York, 13244-4100, USA
Fax: (315) 443-5806
Email: ashay.jawale@gmail.com
Email: jspark@syr.edu
*Corresponding author

Abstract: Today, many stores and users adopt mobile payment services due to the various benefits that the technology can provide. Users can make transactions with their mobile devices such as smart phones instead of physically handing over cash or swiping credit cards. Stores can implement the payment service in a relatively simple and inexpensive way. For both users and stores, the technology increases speed of the check-out process thus reducing the waiting time. The time savings may give more profits to stores. Although the new mobile payment service can provide users and stores with various benefits, it also introduces new security concerns and vulnerabilities. In this paper, we analyse the security features in Apple Pay and discuss possible ways to make it more reliable. Furthermore, once we delve into security vulnerabilities in Apple Pay, we propose the possible solutions along with their implementation to overcome the security concerns in the service.

Keywords: Apple Pay; mobile payment; secure transaction.

Reference to this paper should be made as follows: Jawale, A.S. and Park, J.S. (2018) 'Towards trusted mobile payment services: a security analysis on Apple Pay', *Int. J. Internet of Things and Cyber-Assurance*, Vol. 1, No. 1, pp.76–90.

Biographical notes: Ashay S. Jawale has been working and involved with theoretical/practical research cyber security domain for over four years. He has broad range of work experience varying from working with network security devices, networking devices, implementing security access control and policies, working with AWS cloud computing services and analysing security attacks. He has extensively worked in cloud domain especially with Amazon web services and is currently working with a security firm to analyse security attacks and research about ways to prevent them. In the past, he has worked as a security engineer and as a cloud engineer wherein he was assigned the task of managing the network security of clients' infrastructure, and ensuring security compliance. He is also an independent researcher in security domain and has published his research.

Joon S. Park has been involved with theoretical/practical research and education in cyber security. He is Syracuse University's Principal Contact Faculty at the Center of Academic Excellence (CAE) in Information Assurance/Cyber Defense (IA/CD) and CAE-R (Research), which are

designated by the National Security Agency (NSA) and the Department of Homeland Security (DHS). Since he joined the School of Information Studies (iSchool), he has been the lead faculty member in developing the security curriculum at the iSchool and served as the Founding Director of the Certificate of Advanced Study (CAS) in Information Security Management (ISM).

This paper is a revised and expanded version of a paper entitled 'A security analysis on Apple Pay' presented at The European Intelligence and Security Informatics Conference (EISIC), Uppsala, Sweden, 17–19 August 2016.

1 Introduction

Technology has enabled users to usher the power of hand held devices in every walk of life. This relates to willingness of the audience to use mobile devices for maximum purposes. Today, one of the major mobile applications is the electronic payment service through mobile devices such as cell phones, smart watches, PDAs, and tablets. A mobile payment (also known as mobile commerce) is a new vertical in the commerce. A mobile payment can be defined as a payment where a mobile device is used to initiate, authorise, and confirm an exchange of financial value in return for goods and services (Au and Kauffman, 2008; Karnouskos, 2004; Leavitt, 2010, 2012). Mobile payments are supplement to cash, checks, and cards. The fact that mobile payment can be used for variety of purposes including utility bill payments, online shopping, and transfer of money within accounts makes them more popular. Users can make transactions with their mobile devices such as smart phones instead of physically handing over cash or swiping credit cards. Stores can implement the payment service in a relatively simple and inexpensive way. For both users and stores, the technology increases speed of the check-out process so that the waiting time in line can be reduced. The time savings may give more profits to stores, especially in a busy time period.

Recently, Apple took a step further and introduced its own mobile payment platform also known as Apple Pay, which is a significant addition to the world of mobile payments. It competes with existing mobile payment platforms such as Google Wallet (Camp, 2011; Google Inc., 2016), PayPal (Grabianowski and Crawford, 2005; Poland, 2016), Samsung Pay (Samsung, 2016; Holly, 2015), and other existing mobile payment services (Balan and Ramasubbu, 2009; Boyd, 2005; Michael and Michael, 2015). A mobile payment service should be easy to use, but still secure. In general, in order to use the *Apple Pay* servicer, the user has to add his debit or credit card information into the mobile device that supports *Apple Pay*. The *Apple Pay* management app (i.e., Passbook) then transfers the information to the card issuer (e.g., bank) network to link the payment information with Apple Pay (this process is also known as provisioning). After the process, at the time of transaction the user can make a transaction with the mobile device at Apple Pay supported point of sale (POS) terminals or over the internet. Technically, Apple Pay uses near field communication (NFC) technology (Lumpkins and Joyce, 2015; Bodhani, 2013; Eun et al., 2013) for transactions without sharing the details of the payment cards with merchants. Instead, a random device number is generated and transmitted for the transaction. In this way, the Apple Pay service can increase the level of security (i.e., the protection of card information) during the transactions. Optionally,

Apple Pay can be integrated with biometrics so that the user has to authorise the payments through his fingerprints and/or passcode.

Although the new mobile payment service can provide users with various benefits, it also introduces new security concerns and vulnerabilities (Zhang et al., 2016; Wang et al., 2016; Park and Lando, 2008; Brooks et al., 2015; Hong et al., 2008; Nambiar et al., 2004). In this paper we analyse the security features in the Apple Pay service and dive deep in working of Apple Pay. Further, we discuss the risks and vulnerabilities this system faces and discuss possible ways to make it more secure. We also discuss other payment services briefly and how they operate with reference to Apple Pay. Furthermore, once we delve into security vulnerabilities in Apple Pay we propose the possible solutions along with their implementation to overcome the security concerns in the service.

2 Related work

Today, there are various platforms available for mobile payments. Popular amongst them are PayPal, Google wallet, Alipay, and Samsung Pay. In this section, we discuss the key features of each service and their tradeoffs.

PayPal (Grabianowski and Crawford, 2005; Poland 2016) is a platform to transfer money online between individuals and businesses. It is a subsidiary of eBay Inc. and can be used to shop online as well as to pay bills. *PayPal* enables users to receive money from other PayPal accounts. A unique feature of *PayPal* is that it has facility for its customers to link their credit or debit cards to the *PayPal* accounts and make payments through the card linked. An E-commerce business can use certain *PayPal* features to link their website with the *PayPal* payment options. *PayPal* is an unregulated bank – meaning it does not require following rules and regulations laid out for banks. Therefore, *PayPal* is not liable for loss of amount. It is not required to offer security or dispute resolution services. PayPal is also known to freeze its customers' accounts without any notice thus making that account inoperable until resolved.

Google Wallet (Camp, 2011), (Google Inc., 2016) is a digital service provided by *Google* for electronically managing money. With this service, users can store their payment cards in *Google Wallet*, shop online or in stores, and transfer money for transactions. When paying through *Google Wallet* in stores, users have to hold their phones near POS terminals to finish checkout. This is achieved by using NFC technology with a mobile device such as a smartphone and POS terminal that support the *Google Wallet* service. There are a few extra procedures required to use the service. The user needs to wake up his smartphones if locked and enter a PIN for preceding certain transaction. *Google Wallet* supports payments from most of the credit or debit card providers. One of the unique features of this service is that users can view their available balances, which may eliminate the need to store cards on smartphones. Google wallet offers 24 × 7 fraud monitoring for transactions in the USA, wallet PIN to authorise transactions, and remote wipe. Its adoption rate was slow, partly due to the lack of support from carriers. *Google* stores all the details about the users' accounts and payment history, which makes account details more susceptible in case the servers are hacked or information is leaked.

Samsung Pay (Samsung, 2016; Holly, 2015) uses both near-field communication (NFC) and magnetic secure transmission (MST) technologies. *Samsung Pay* enables

mobile payments through commonly found tap-and-pay terminals using NFC. In cases where only older terminals are available, Samsung Pay can use MST to emulate a swipe made by a traditional magnetic stripe credit card. This technology is fairly easy and only requires the user to tap his smartphone on the side of the POS terminal where a card is usually swiped. By doing this, MST transmits the data magnetically causing the smartphone to send payment credentials. There are no technological changes required from vendor side to support this. This gives Samsung pay access to much more POS terminals that are not equipped with NFC technology yet.

Alipay (Nowlin, 2014) service is provided by *Alibaba* and becomes one of the major Chinese mobile payment platforms. What makes this service more popular than others is the buyer protection feature it provides to its customers. Just like *PayPal*, once customers make their payment, *Alipay* holds that payment with itself until satisfactory confirmation of received product is given by buyer. Payment is released to the seller after this confirmation. In other words, this service acts like an escrow service. Similar to other mobile payment platforms, users can shop online, in stores, transact, pay bills, and additionally invest money in stocks. *Alipay* enables *Alibaba* sellers and buyers to decrease their business risks since the *Alipay* system controls the whole transaction process. On downside, accumulated data of the Chinese customers may prove to be inapplicable to the European or American markets due to cultural differences *Alipay* also stores customer data and their buying trends on its servers making it susceptible to leak.

3 Security analysis on Apple Pay

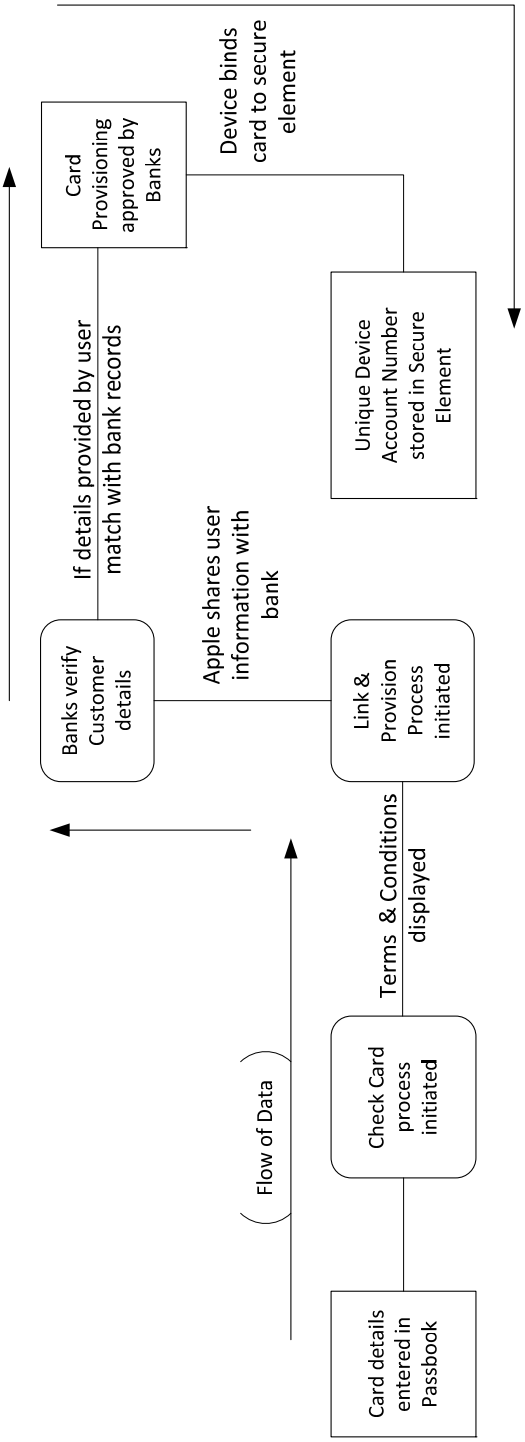
Apple Pay has a relatively simple setup procedure and easy user interface. It is designed to protect user's personal information. We analyse the main components and process of Apple Pay in this section.

3.1 Key components

Secure Element (SE): the SE is an industry-standard, certified chip running the Java Card platform, which is compliant with financial industry requirements for electronic payments (PCI-DSS). It is specifically designed to manage the Apple Pay service and includes payment applets certified by payment networks. Only the payment networks and payment applet's security domain know the cryptographic keys to encrypt/decrypt the payment card information. During a transaction at merchant's location, the POS terminal communicates directly with SE in user's mobile device such as *iPhone* using the NFC controller (Apple Inc, 2015).

NFC controller: The NFC controller acts as a gateway to SE. It ensures all payments are contactless and handles NFC protocols. Payments are termed as contactless transactions only when they come from an infield POS terminal. Every payment has to be authorised using *touch ID* or *passcode* in the mobile device via *secure enclave*. After authorisation, the payment applet within SE prepares contactless responses which are directed into NFC field by this controller. This increases the security in transactions as payment authorisation details are confined only within NFC field. When paying through apps, payment authorisation details are encrypted by SE to the Apple server and then sent out via an application processor.

Figure 1 Card provisioning process in Apple Pay



3.2 Card provisioning

When user adds a payment card to *passbook* in an *Apple* device, it should securely send this information along with other information such as user's account and device ID to the corresponding bank. *Apple Pay* uses two server side calls to send and receive the service data from/to banks for card provisioning, check card and link and provision. Banks use these calls to verify, approve, and add cards to *Apple Pay*. This client-server communication is SSL encrypted. Card numbers and details are not stored on the device nor on *Apple* servers. Figure 1 depicts the card provisioning process in the Apple Pay service. In the provisioning process a unique device account number is created, encrypted, and stored in SE, which is isolated from iOS. *Apple* or its servers cannot access this number. Later, more cards can be added into Apple Pay either manually or through the user's *iTunes* store accounts (Apple Inc, 2015).

In the manual procedure, the details on the card are used to facilitate the provisioning process. The user has to enter the information in *passbook*. After this, *check card* process verifies the card number and expiration details that are then encrypted and sent to *Apple Pay* servers. *Terms and conditions* of the corresponding bank is returned with the *check card* process and displayed for the user. Once user accepts it, *Apple* sends the ID of terms and conditions that were accepted as well as the card verification value (CVV) number to the link and provision process. *Apple* shares the information from the device with the issuing bank or network (e.g., the last four digits of the phone number, the device name, and the location coordinates of the device at the time of provisioning, and so on). Using this information, the issuing bank will determine whether to approve adding the card to *Apple Pay*. The *link and provision* process is initiated and the device begins to download the *Passbook* pass file representing the credit or debit card and the device begins to bind the card to SE.

For adding cards through an *iTunes* account, the user is required to re-enter his *Apple* ID password. The card number is retrieved from *iTunes* and the *check card* process is initiated. If the card is eligible for *Apple Pay*, the similar procedure occurs when the user adds card manually to the passbook. There are additional verification steps for *iTunes* account cards on the file, but those are at sole discretion of the card issuing bank.

3.3 Payment authorisation

SE allows payment to be made only after it receives authorisation from secure enclave, which manages the authentication process and enables a payment transaction to proceed. The default method of authentication is *touch ID* and *passcode* is required after five unsuccessful attempts. Communications between the secure enclave and SE take place over a serial interface, with SE connected to the NFC controller, which in turn is connected to the application processor. Even though not directly connected, secure enclave and SE can communicate via a protected channel using a shared pairing key that is provisioned during the manufacturing process. The pairing key is generated inside secure enclave from its UID key and SE's unique identifier. The pairing key is then transferred from secure enclave to a hardware security module (HSM) in the factory, which has the key material required to then inject the pairing key into SE. When the user authorises a transaction, Secure Enclave sends signed data about the type of authentication and details about the type of transaction (contactless or within apps) to SE, tied to an authorisation random (AR) value. The encryption and authentication of the

communication is based on advanced encryption standard (AES) (Apple Inc, 2015; Federal Information Processing Standards Publication 197, 2001). Figure 2 describes the general *Apple Pay* payment process in stores.

The AR is generated in secure enclave when a user first provisions a credit card and is persisted while *Apple Pay* is enabled, protected by Secure Enclave's encryption and anti-rollback mechanism. It is delivered to the SE via the pairing key and changes when new card is provisioned. Using the pairing key and its copy of the current AR value, SE verifies the authorisation received from secure enclave before enabling the payment applet for a contactless payment (Apple Inc, 2015).

1 *Transaction-specific dynamic security code:* all payment transactions originating from the payment applets include a transaction specific dynamic security code along with a device account number. This is a onetime code that is calculated using a counter, which is incremented for each new transaction, and a key, which is provisioned in the payment applet known to the respective bank. Some other data may also be used in calculation of these codes depending on the payment schema (e.g., a random number generated by the payment applet, another random number generated by the terminal – in the case of an NFC transaction, and another random number generated by the server – in the case of transactions within apps). These payment codes are provided to the respective bank's network that allows them to verify each transaction.

2 *Contacless payments:* whenever an Apple-Pay-ready device detects an NFC field, it will present the user with default the credit/debit card. Then the user has to follow the authentication process using touch ID or passcode for payment to be done. Once the user authenticated, the device account number and a transaction-specific dynamic security code are used for processing the payment. Neither Apple nor user's device send the full actual credit or debit card numbers to merchants.

3 *Payment within connected applications:* when paying though apps, Apple servers receive encrypted transaction information. This information is re-encrypted with the merchant-specific key and then forwarded to the merchant. Apple Pay retains approximate purchase amount, but never retains what customer is buying. When an app requests a payment, it calls an API to confirm if device supports Apple Pay and if the user has a card provisioned in passbook that can make payments. The app then requests information required to fulfil the transaction such as billing address, contact information, and so on. The app then asks iOS to present Apple Pay sheet. At this time, address and zip code are presented to the app to determine shipping costs and taxes. The user has to authorise the payment using his pass code or though ID and only then entire information in Apple Pay sheet is forwarded to merchant. After authorisation, a connection is initiated to Apple servers to obtain a cryptographic nonce. The nonce, along with other transaction data, is passed to SE to generate a payment credential that will be encrypted with an Apple key. After this, encrypted payment credential from SE is passed to Apple Pay servers which verify the nonce and re-encrypt the payment credentials with merchant's key. It is then returned to the device, which hands it back to the app via the API. App then passes it to merchant's system for processing. The merchant can then decrypt the payment credential using its own private key. This information along with signature from Apple servers allows the merchant to verify that the transaction was intended for this specific merchant.

4 *Deleting card information:* there are several ways for a user to do this operation by placing the device in the lost mode, using Find My iPhone, removing cards through iCloud settings, or deleting directly through Passbook. Cards are also deleted when users restore their using recovery mode or wipe out their devices using Find my iPhone.

4 Security concerns in Apple Pay

Apple Pay provides security mechanisms to secure the transactions and payment data. To boost it further, *Apple* and the compliant devices do not store any card information which could be tied back to the user. Instead, the service uses a unique device account number, which is totally different than credit/debit card number. This device account number along with the AR value and private security keys makes it possible for *Apple Pay* to do transactions in a secure manner.

However, there are certain design issues that need to be taken into consideration. *iOS* has some vulnerabilities which – though *Apple* patches these from time to time – can be exploited by hackers and may pose a threat for *Apple Pay* itself. We discuss the security concerns in the current *Apple Pay* service in this section.

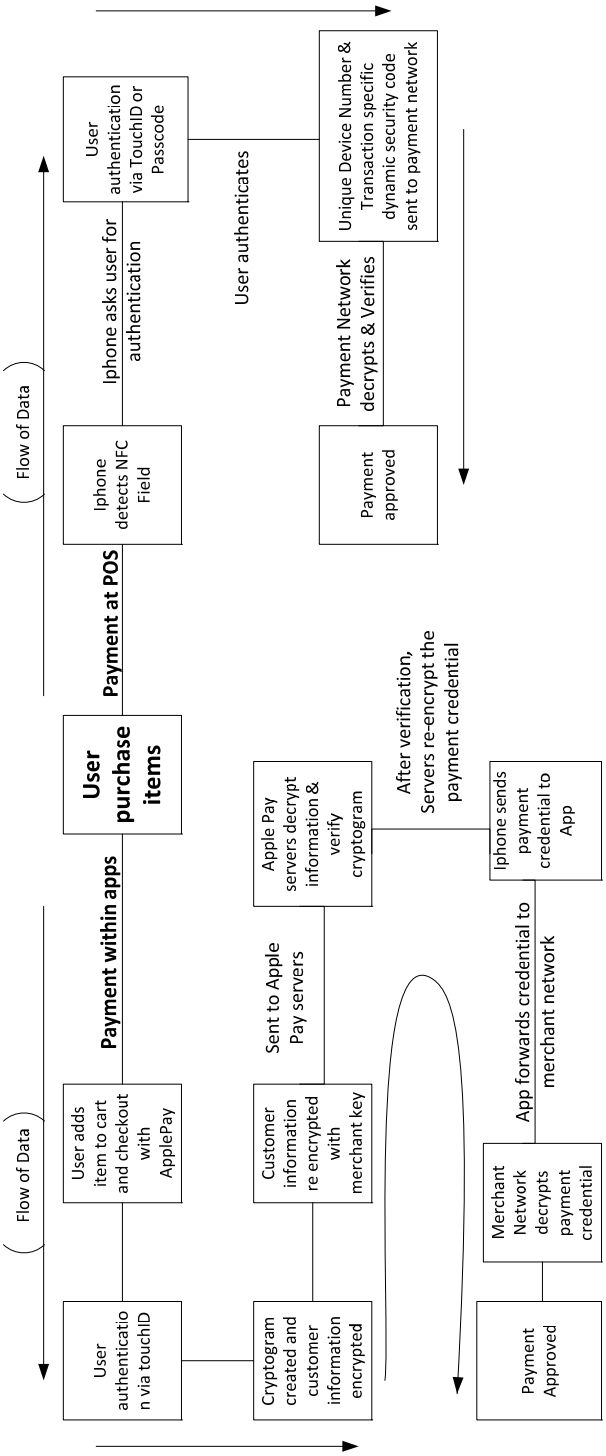
4.1 Security concerns with Touch ID

Apple Pay uses *Touch ID* with fingerprints for authentication. While biometrics is still an emerging technology, the authentication process can be bypassed with extra efforts. Actually, it has been demonstrated how *iPhone5S Touch ID* can be bypassed by creating a fake fingerprint (Ratha et al., 2015; Frank, 2013). Humans leave behind their fingerprints various places. The attack showed how a fake fingerprint ID can be produced from user's original fingerprint left behind on a glass. Therefore, if a user loses his phone or gives it to someone for using it (someone who has malicious intent) the attacker can create a fake fingerprint ID of the original user by obtaining user's fingerprints on the phone. In this way, the attacker can easily unlock, access the device, and use *Apple Pay* for buying items through apps.

4.2 Threats from 3rd party applications

In another case, third party apps or someone may load malicious software or a key logger in the legitimate user's Apple-Pay-ready device, which can track user's inputs or data (Cox, 2014). There are two main ways to enter card details into *Apple Pay*'s passbook: either by manually typing the card information in the passbook or by taking the picture of the card using the built-in camera of the device. Even though the device and *Apple* Server's do not store this information or card image, a key logger may attempt to record the keystrokes used while entering card information or malicious application may capture the image used for entering card details. This information can be later on misused by the attacker.

Figure 2 *Apple pay* payment process in store



4.3 Weakness in card provisioning

Since the real card information is never transmitted when transactions are done via *Apple Pay*, the card information cannot be hijacked when transaction is being done and data is being transferred to a POS terminal. On the contrary, vulnerability exists in the earlier part, when debit/credit card is added into passbook. Fraudsters may add stolen credit/debit card information in *Apple Pay* and provision it. According to *Apple*'s rules, it is credit/debit card issuing bank's responsibility to verify the legitimacy of cards when they are added into *Apple Pay*. Apple does not take any responsibility for verification of card. The fraudsters, who have access to credit card information also, may have enough information about user's personal information such as date of birth, SSN, address, or zip code. As such this personal information can be easily obtained by running a Google search or browsing through social media. Interesting fact to note here is that when you are entering card details in Apple Pay, there is no mechanism to shut out a user of the system after number of incorrect retries. One piece of information which maybe unavailable to the fraudsters would be 3-digit security code (CVV code or CVV or CSC code) and they may brute force the Apple Pay to try various combinations – 3-digit code will give rise to only 999 unique codes. Apple Pay does display an error saying, 'Could not connect to Apple Pay, Apple Pay is temporarily unavailable'. However, there are ways to get around this error by turning off the passcode and then adding the card again. The Wallet (Apple Pay) will prompt to re-enable the passcode and accept the card if the details match.

4.4 Poor card verification policies

A recent informal study by anti-fraud firm Pindrop indicates that card verification varies widely from issuer to issuer (Fox-Brewster, 2016). The study involved provisioning four different issuer cards in the Apple Pay system and verification of them was analysed. One of the provider did perform a very low level check that if the name of the card matched with the name on the Apple Pay account. It is relatively easy for a fraudster to change the name of the device and get the card provisioned. Another card issuer merely accepted the numbers on the card and it was provisioned. There were no checks performed to verify the identity. One of the providers did require the researcher to call and verify the card holder's details. But the researcher was easily able to get past this using social media and Google to this advantage. The last provider required a call back and entering a random number to validate the card provisioning (Fox-Brewster, 2016; PYMNTS, 2016). As such, it depends on verification policies of the card issuing organisation. While some banks do ask the users to make verification calls to provision their cards into *Apple Pay, others do not thus making the system vulnerable to exploit*. When these calls are made by a fraudster, they often have enough personal information about the real card-owner to answer those verification questions. As such, the system hits its break point in such a situation. After verification, the fraudster can shop online or at a POS terminal in stores, and the fraud will go undetected for a long period since all the information is already verified. Banks also try to make verification steps simple since they do not want to anger or frustrate their legitimate customers (Soni, 2010). Actually, there are websites and services that offer stolen credit card information at a very cheap price. Some underground document forgery websites and services like 'carding' sites or dark web are sources of such information. As such, it should be noted that this is not vulnerability in the Apple's

iOS but could be better described as a double edged sword which both protects customers' card information as well as is the weakest link in the Apple Pay system thereby easiest path for the exploiters to exploit the system.

5 Recommendations and reinforced apple pay security model

It may appear that banks may be blamed for the most part but Apple is not completely faultless as the firm's platform is becoming an easy way to effectively use someone else's card. In this case, *Apple* should have taken the responsibility in verification of identity and card details to ensure that this type fraud is kept at bay.

5.1 Rate limiting mechanism

Apple should introduce 'rate limiting' mechanism to prevent exploiters from brute forcing the Apple Pay with incorrect information like CVV code (Fox-Brewster, 2016). This type of mechanism is present in Apple Pay's competitor systems like Samsung Pay and Google Wallet (PYMNTS, 2016). This gives them an upper hand in card verification process compared to Apple Pay. Rate limiting mechanism can be as simple as shut out the users from the Apple Pay after three incorrect tries and wait for few hours like 24 hours before they could retry adding another card. This will deter the crooks from retrying multiple times to an extent. The rate limiting mechanism may also factor that use of Apple Pay should not be affected but only adding of that particular card. Apple should act pro-actively and be much more diligent about preventing such automated brute force attacks rather than relying on the banks and financial institutions to catch the frauds.

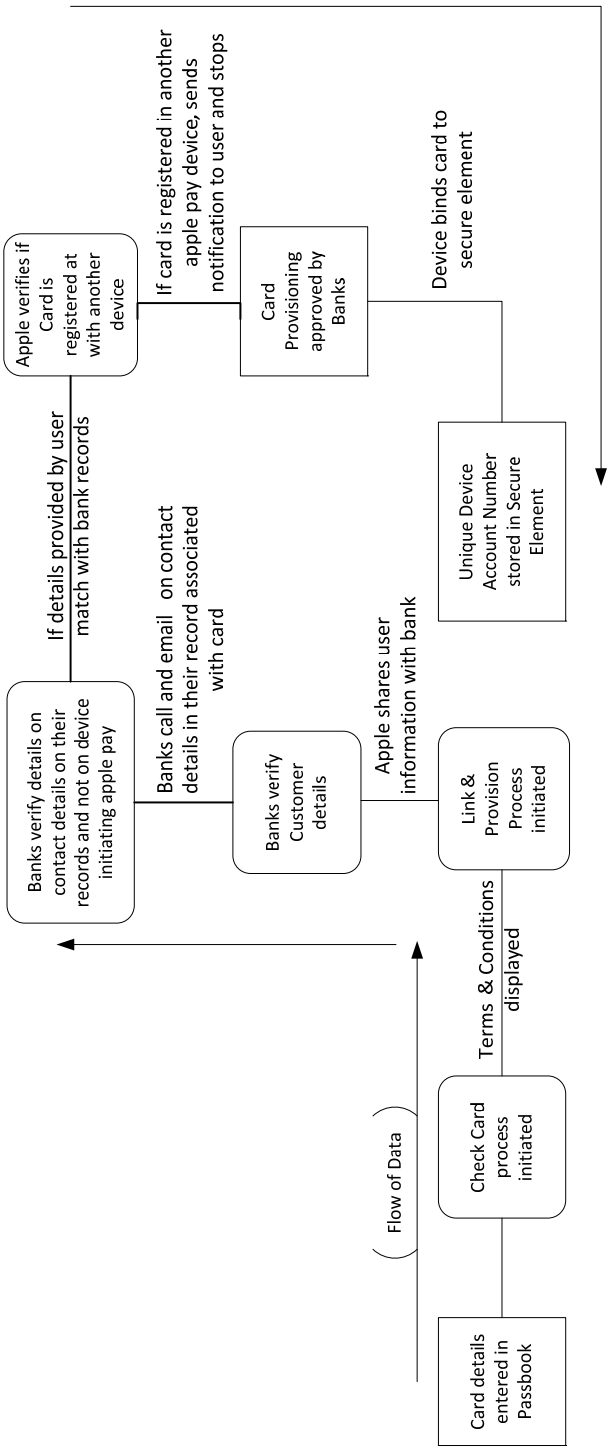
5.2 Limited service delegation and limited cards in Apple Pay

In terms of the service delegation, because Apple Pay uses fingerprint authorisation to release payment transactions a user cannot hand over his *Apple-Pay*-ready device to other people to have them make a transaction on behalf of the owner (i.e., no delegation service). Furthermore, when the device runs out of charge and if the user does not have any other card or cash in his wallet, the user may easily find themselves broke (Irwin, 2014).

Apple should also make a norm which makes it mandatory to make one credit card or at the most two active per *Apple Pay*. Currently, *Apple Pay* allows multiple cards registered within it. This might be more convenient for the user, but it also increases the level of vulnerability based on the security principle of least privilege. The users should be able to easily delink a particular card when users decided to suspend or erase the information from Apple Pay or when fraud is reported.

For a more trusted *Apple Pay* service, provisioning of cards should be shared responsibility of banks and *Apple*. Whenever a card is added into *Apple Pay*, both the entities should verify legitimacy on individual basis.

Figure 3 Reinforced *Apple Pay* card provisioning process



5.3 Bank's role

Banks often try to keep it simple for their customers regarding verification calls. As such, banks are exposed to risks explained above. Banks should have stricter verification rules. They can initiate verification process over an email and registered phone number associated with the credit card. Answering personal questions such as SSN, date of birth, or zip code through registered phone number and email will add an additional layer of security. Since chances of both – registered phone and email – getting compromised along with credit card information are very rare, this can tackle the issue of legitimacy. The proposed card provisioning process that can make the current *Apple Pay* service more reliable is depicted in Figure 3.

Banks can also display information about credit card being linked up with which Apple Pay device in the Internet banking account of an individual. In this case, if stolen credit card information is registered with an *Apple Pay* application as well as verified, it will be visible to the real owner of credit card in their Internet banking account. Therefore, even if a fraudster tries to register the stolen card information in Apple Pay, they cannot use it to make any transaction, because the account can be delinked with the Apple Pay device when the real owner receives a fraud notification of credit card and avert.

6 Conclusions and future work

Today, mobile payment services become more popular to both stores and users due to the various benefits that they can provide. Recently, *Apple* took a step further and introduced its own mobile payment platform also known as Apple Pay, which is a significant addition to the world of mobile payments. It competes with existing mobile payment platforms such as Google Wallet (Camp, 2011; Google Inc, 2016), PayPal (Grabianowski and Crawford, 2005; Poland 2016), Samsung Pay (Samsung, 2016; Holly, 2015), and other existing mobile payment services. Although the new mobile payment service can provide users and stores with various benefits, it also introduces new security concerns and vulnerabilities. In this paper we analysed the security features in the *Apple Pay* service and discuss possible ways to make it more reliable. Furthermore, once we delved into security vulnerabilities in *Apple Pay* we proposed the possible solutions along with their implementation to overcome the security concerns in the service. The paper contributes towards to creating a base model for mobile payment systems focusing on vulnerabilities present in Apple Pay. As such these vulnerabilities are combination of poor policies and implementation. This paper identifies them and recommends ways to overcome those. This helps create a standard baseline for existing and future mobile payment systems. Stronger verification techniques will deter fraudulent use of cards and rate limiting mechanisms will deter crooks from trying stolen card details. This way, it helps to build a stronger reinforced security model, which would be difficult to break and thereby secure financial aspect associated with them. Security in Apple Pay can be better described as a double-edged sword, performing an excellent job of protecting customer's card data from fraudsters who try to use misappropriated cards.

In the future, we are planning to conduct an in-depth analysis on each of other existing mobile payment services including Google Wallet, PayPal, and Samsung Pay, and compare the results across the different services. We also plan to evaluate and

modify our analysis as these services evolve and face new threats. We intend to create a base security model for mobile payment services that addresses vulnerabilities across all these services. We believe the comprehensive outcomes of this research will make a significantly positive impact on the trusted mobile services in the future.

References

Apple Inc. (2015) *iOS Security*, Apple.com [online] https://www.apple.com/business/docs/iOS_Security_Guide.pdf (accessed 20 April 2015).

Au, Y.A. and Kauffman, R.J. (2008) 'The economics of mobile payments: understanding stakeholder issues for an emerging financial technology application', *Electron. Commer. Rec. Appl.*, July, Vol. 7, No. 2, pp.141–164.

Balan, R.K. and Ramasubbu, N. (2009) 'The digital wallet: opportunities and prototypes', in *Computer*, April, Vol. 42, No. 4, pp.100–102, DOI: 10.1109/MC.2009.134.

Bodhani, A. (2013) 'New ways to pay [communications near field]', in *Engineering and Technology*, August, Vol. 8, No. 7, pp.32–35, DOI: 10.1049/et.2013.0716.

Boyd, J. (2005) 'Here comes the wallet phone [wireless credit card]', in *IEEE Spectrum*, November, Vol. 42, No. 11, pp.12–14, DOI: 10.1109/MSPEC.2005.1526896.

Brooks, T., Shoniregun, C. and Park, J.S. (2015) 'Cyber-assurance through embedded security for the internet of things (IoT)', *Instituting Cyber-Assurance: Information Assurance for the Internet of Things*, Wiley-IEEE.

Camp, J.V. (2011) *'How Google Wallet Works'* | *Digital Trends*, 20 September [online] http://www.digitaltrends.com/how-to/how-google-wallet-works/ (accessed 12 February 2015).

Cox, J. (2014) *'Apple reveals unprecedented details in iOS security'* | *Network World*, 6 March [online] http://www.networkworld.com/article/2174973/smartphones/apple-reveals-unprecedented-details-in-ios-security.html (accessed 18 October 2015).

Eun, H., Lee, H. and Oh, H. (2013) 'Conditional privacy preserving security protocol for NFC applications', in *IEEE Transactions on Consumer Electronics*, February, Vol. 59, No. 1, pp.153–160, DOI: 10.1109/TCE.2013.6490254.

Federal Information Processing Standards Publication 197 (2001) *Advanced Encryption Standard (AES)*, November.

Frank (2013) *'Chaos Computer Club breaks Apple TouchID'* | *Chaos Computer Club*, 21 September [online] http://www.ccc.de/updates/2013/ccc-breaks-apple-touchid (accessed 12 October 2015).

Google Inc. (2016) *Google Wallet*, Google US [online] https://www.google.com/wallet/ (accessed 13 February 2015).

Grabianowski, E. and Crawford, S. (2005) *'How PayPal Works'* | *HowStuffWorks.com*, 13 December [online] http://money.howstuffworks.com/paypal.htm (accessed 21 March 2015).

Holly, R. (2015) '*Samsung Pay on the Galaxy S6, and why it matters'* | *Android Central*, 10 March [online] http://www.androidcentral.com/forums-samsung-pay-galaxy-s6-and-why-it-matters (accessed 24 May 2015).

Hong, D., Runtong, Z. and Tian, L. (2008) 'Risk assessment management for mobile payment security', *Service Operations and Logistics, and Informatics, 2008, IEEE/SOLI 2008. IEEE International Conference on*, pp.1966–1970, Beijing, DOI: 10.1109/SOLI.2008.4682854.

Irwin, N. (2014) *'Apple Pay Tries to Solve a Problem That Really Isn't a Problem'* | *New York Times*, 10 September [online] http://www.nytimes.com/2014/09/11/upshot/apple-pay-tries-to-solve-a-problem-that-really-isnt-a-problem.html?_r=2 (accessed 21 October 2015).

Karnouskos, S. (2004) 'Mobile payment: a journey through existing procedures and standardization initiatives', in *IEEE Communications Surveys and Tutorials*, Fourth Quarter, Vol. 6, No. 4, pp.44–66.

Leavitt, N. (2010) 'Payment applications make e-commerce mobile', in *Computer*, December, Vol. 43, No. 12, pp.19–22, DOI: 10.1109/MC.2010.357.

Leavitt, N. (2012) 'Are mobile payments ready to cash in yet?', in *Computer*, September, Vol. 45, No. 9, pp.15–18, DOI: 10.1109/MC.2012.298.

Lumpkins, W. and Joyce, M. (2015) 'Near-field communication: it pays: mobile payment systems explained and explored.', in *IEEE Consumer Electronics Magazine*, April, Vol. 4, No. 2, pp.49–53, DOI: 10.1109/MCE.2015.2395011.

Michael, K. and Michael, M.G. (2015) 'Apple watch temptation: just visit the app store', in *IEEE Consumer Electronics Magazine*, October, Vol. 4, No. 4, pp.120–122.

Nambiar, S., Lu, C.T. and Liang, L.R. (2004) 'Analysis of payment transaction security in mobile commerce', *Information Reuse and Integration, IRI 2004, Proceedings of the 2004 IEEE International Conference on*, pp.475–480, DOI: 10.1109/IRI.2004.1431506.

Nowlin, J. (2014) *'What You Need to Know About the Chinese Consumer: Understanding Alipay' | ChannelAdvisor*, 19 September [online] http://www.channeladvisor.com/blog/?pn=marketplaces/what-you-need-to-know-about-the-chinese-consumer-understanding-alipay (accessed 28 May 2015).

Park, J.S. and Lando, J. (2008) 'E-commerce: the benefits, security risks, and countermeasures', in Gupta, J. and Sharma, S. (Eds.): *Handbook of Research on Information Security and Assurance*, pp.7–17, IDEA Group Publishing.

Poland, A. (2016) *'Advantages and Disadvantages of Using PayPal' | Chron.com* [online] http://smallbusiness.chron.com/advantages-disadvantages-using-paypal-55073.html (accessed 27 March 2015).

PYMNTS (2016) *'Apple Pay's Low-Tech Security Problem' | PYMNTS.com*, 4 March [online] http://www.pymnts.com/apple-pay-tracker/2016/apple-pays-low-tech-security-problem/ (accessed 31 October 2016).

Ratha, N.K., Connell, J.H. and Pankanti, S. (2015) 'Big data approach to biometric-based identity analytics', in *IBM Journal of Research and Development*, March–May, Vol. 59, Nos. 2/3, pp.4:1–4:11, DOI: 10.1147/JRD.2015.2394514.

Samsung (2016) *Samsung Pay*, Samsung Electronics America [online] http://www.samsung.com/us/samsung-pay/ (accessed 21 May 2015).

Soni, P. (2010) 'M-payment between banks using SMS [point of view]', in *Proceedings of the IEEE*, June, Vol. 98, No. 6, pp.903–905, DOI: 10.1109/JPROC.2010.2047216.

Fox-Brewster, T. (2016) *'Here's Proof Apple Pay Is Useful For Stealing People's Money' | Forbes Welcome*, 1 March [online] http://www.forbes.com/sites/thomasbrewster/2016/03/01/apple-pay-fraud-test/#3967d7203c15 (accessed 22 October 2016).

Wang, Y., Hahn, C. and Sutrave, K. (2016) 'Mobile payment security, threats, and challenges', *2016 Second International Conference on Mobile and Secure Services (MobiSecServ)*, pp.1–5, Gainesville, FL, DOI: 10.1109/MOBISECSERV.2016.7440226.

Zhang, Z., Wang, X. and Sun, L. (2016) 'Mobile payment anomaly detection mechanism based on information entropy', in *IET Networks*, Vol. 5, No. 1, pp.1–7, DOI: 10.1049/iet-net.2014.0101.