SECURE DIGITAL WALLET AUTHENTICATION PROTOCOL

by

Kardam Tiwari

Submitted in partial fulfilment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
April 2016

Dedicated to my Parents and Chavi,

Table of Contents

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Digital wallets have gained popularity for secure storage of credit cards. They are time saving, secure and track the expenditure. However, with the high dependency of these wallets on client side devices, the risk of data loss and subsequent financial losses due to physical attacks on the device remain high. Server-side wallets, such as PayPal, are secure behind firewalls and are accessible only through valid usernames and passwords. Due to this issue, client-side wallets are vulnerable, thus impacting their popularity and their widespread use. We propose a new wallet authentication scheme that protects mobile digital wallets against physical attacks. Our scheme uses biometric (fingerprint) authentication without actually storing the fingerprint data on the digital wallet, in addition to using hash chaining and dynamic key generation to ensure that the communication between servers and the clients remains mutually authenticated. The prototype has been implemented in hardware and validated through security analysis.

LIST OF ABBREVIATIONS AND SYMBOLS USED

Ack	Acknowledgement
Dkspub	Decrypted with public key of server
Dkspriv	Decrypted with private key of server
Dkwpriv	Decrypted with private key of wallet
Dkwpub	Decrypted with public key of wallet
DOC	Document
Ekspub	Encrypted with public key of server
Ekspriv	Encrypted with private key of server
Ekwpriv	Encrypted with private key of wallet
Ekwpub	Encrypted with public key of wallet
Fp	Fingerprint
H(fp)	Hash of the Fingerprint
Kspriv	Private Key of Server
Kspub	Public Key of Server
Kwpriv	Private Key of Wallet
Kwpub	Public Key of Wallet
Ktxn	Transaction key
PRNG	Pseudo Random Number Generator
SignS	Signature of Server
SignW	Signature of Wallet
SignW Txn	Signature of Wallet for Transaction
BI	Biometric Identification
DoS	Denial of Service
IDE	Integrated Development Environment
IP	Internet Protocol
POC	Proof of Concept
POW	Proof of Work
SWAP	Secure Wallet Authentication Protocol
TCP	Transmission Control Protocol
USB	Universal Serial Bus
SHA 256	Secure Hash Algorithm
TLS	Transport Layer Security
SSL	Secure Socket Layer
AES	Advanced Encryption Standard
RSA	Rivest Shamir Adleman
MITM	Man in the Middle

WI-FI	Wireless Fidelity
AC	Alternating Current
DC	Direct Current

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my supervisor Dr. Srinivas Sampalli for his guidance and support to complete this research. My appreciation extends to my co-supervisor Dr. Musfiq Rahman for guiding me at every point and making me go for the right turns along the road. Their immense patience throughout the research has been encouraging me to go for an extra mile.

I would also like to thank my parents, family for encouraging me throughout the journey. A special thanks to Raghav, Saurabh, Afiz, Abu, Nitya and Jacki for always pointing out my flaws and helping me in making my idea concrete and real.

Last but not at all the least, special thanks to God for showering his blessings on me.

Chapter 1: Introduction

1.1 Cryptocurrency

Cryptocurrency is a form of digital currency, which is based on cryptography i.e., the art of writing or solving codes. Crypto is a Greek word which means 'hidden' and currency is the accepted form of money issued by the Government. Thus, cryptocurrency is an encrypted form of money. Each country has its own currency with its own exchange rate. Due to this difference in exchange rates, people may have to pay fees for making a wire transfer or making transactions required for international trade. Although the use of cryptocurrency involves a transaction fee, there is no additional charge for international transactions. Cryptocurrency is also safe, encrypted and there is no physical existence. This is one of the major reasons behind the shift from physical currency to electronic currency [1].

1.2 Case study on Bitcoins

Although cryptocurrency is a digital form of money, it still needs to be managed and monitored by centralized authorities such as banks, financial institutions etc. In 2008, Satoshi Nakamoto proposed a new cryptocurrency called bitcoin [2], which as of now is a fully decentralized system. In other words, the bitcoin system is a peer-to-peer non-centralized system. Due to its wide acceptance and lack of centralized control, bitcoins have become the most preferred cryptocurrency.

Bitcoins do not have any hidden charges for using them. The transaction is faster through bitcoin system compared to wire transfer via banks. There is no service or external fee for

a transaction made over bitcoin system from one country to another. The money is received within minutes in a more secure way than traditional cash via bank transfer. Bitcoins are computed over a mathematical function (hashing) without the help of a third party, so all the proofs of transactions are provided to merchants. One of the most important benefits of bitcoin [1] is absolute anonymity. This type of cryptocurrency is favorable over other digital currency.

1.3 Motivation

While the use of bitcoins is becoming widespread, it has several security issues. Bitcoins have been a victim of double spending. The decentralized nature of bitcoins is an asset as well as a curse because not many people believe in this type of architecture (figure 1.1). Customers prefer a third party authentication i.e. an endorsement by a financial institution or by a centralized banking system. Storing bitcoins is a dicey job. There are many types of wallets which can store bitcoins, but none of them guarantee full security. This is also one of the reasons which make bitcoins vulnerable to attack by hackers. In this thesis, we have tried to address this problem and provided a solution for it. For the solution, we have used the combination of biometric, hash chaining and dynamic key generation.

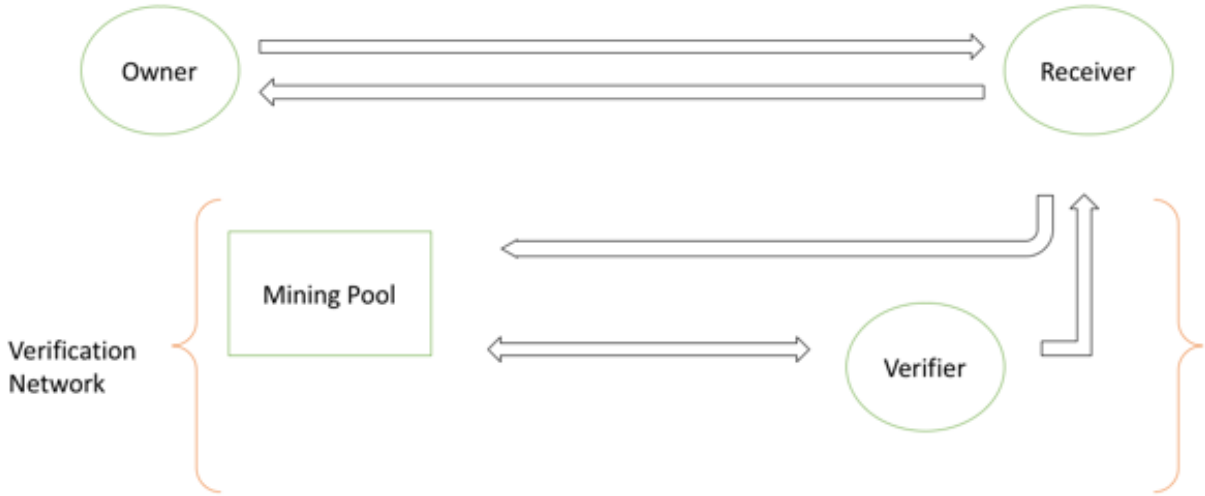

Figure 1. 1 Bitcoin System Architecture

Chapter 2: Literature Review

2.1 An overview of the bitcoin transaction

When a buyer makes a purchase for an item through bitcoins, the seller takes it and sends it to verification process. The verification is done by bitcoin miners. The miners are given incentives for their services. To earn the incentives, the bitcoins miners show a proof of work of mathematical computations in order to verify the bitcoin. The whole process takes less than ten minutes and also confirms if the coin is double spent [2].

Bitcoin is a peer-peer distributed time-stamp network which simply put, is the chain of digital signatures. Time-stamp proves that all the transactions occurring at a particular time are taken in the record. In this architecture, the peer can verify the ownership of a coin. The validity of the coin cannot be verified by the peer, so the authentication of a bitcoin is done by the bitcoin miners. Peer refers to the users of a bitcoin, in daily life, they are sellers and buyers. The computational proof of the chronological order of transactions leads to the elimination of the third party trust.

2.2 A detailed view of the process involved in bitcoin

Bitcoin is sent over a bitcoin network. The bitcoin network is responsible for keeping track of all the transactions going on in the network. It is the work of miners to control the transaction and create records on a common ledger on the database of a bitcoin network. Keeping track of transactions during a set period of a list is called a ‘block’. Then these particular blocks together form a chain called a block chain [2] [3] [4].

This includes an important step of making a hash. Since a block chain has the record of all the transactions made on the bitcoin network, it is used to explore any transaction made between any bitcoin address at any point on the network. Every time a new transaction block is found on the network, it is added to the blockchain.

2.3 Peer-to-Peer Verification Network

The data stored is secured and protected through hashing. The transactions in the block are used to generate a hash with the help of additional data (which is extracted by the miners) [2]. They comprise of a hash of the last block stored in the chain because this confirms that the block occurring after this is legitimate and any meddle with the block will be noticed. let's assume: a hacker tries to fake a transaction by changing a block that has already been stored in a block chain, then the hash would change. The block's authenticity is checked by running a hash function on it. In case, the miners see that the hash is different this time then the block is immediately discarded. This stops double spending and also makes the peer-to-peer network more reliable. Tampering with one block will automatically result in the wrong calculation of all the following blocks. Although the method sounds easy, getting a block is a tedious task for the miners, miners seal off a block, they will compete with each other to do this using a software specifically assigned to mine blocks. Hash is easy to produce but it is the bitcoin network which has to make it more difficult. Otherwise anyone can hash any number of transaction and with the help of this, any number of bitcoin can be mined.

Now, here lies the beauty of bitcoin network. They introduce the proof of work, so that whoever joins the network and tries to mine the bitcoin should also have a proof of work

that has been done. Bitcoin protocol would not accept any old hash. A block's hash has to look a certain way;

- The number of zeroes at the beginning of the hash should be equal the value of the nonce.
- Miners cannot meddle with the transaction data in the block. But they must change the data they are using to create a different hash. This is the concept of a nonce.

The nonce is used with the transaction data to create the hash. If the hash does not fit the required format, its nonce is changed. Until the time, the required nonce is derived and it may take several attempts to get the desired output [2] [5].

Category	Name of the attack	Victim
Wallet Theft	Stoneman loss, Stefan Thomas loss, bitomat.pl loss	Stoneman, Stefan Thomas, Bitomat.pl
Ownership Theft	June 11 Mt. Gox incident, mybitcoin theft	Mt. Gox, mybitcoin users
Fraud	Tony Silk road scam	Buyers on silk road

Table 2. 1 List of few attacks

2.4 General attacks on bitcoins

Since bitcoin is the most popular crypto currency, it is vulnerable to attacks. It has been targeted several times in the past [6] [7]. The attacks are of various types: wallet theft, anonymity, double spending, packet sniffing and denial of service attacks. The flowchart of attacks has been mentioned in figure 2.1.

2.4.1 Wallet theft

Wallet attack is the first kind of attack that was done on a bitcoin network. There are three types of wallets:

- **Online wallet**: All the coins owned by bitcoin users are saved online along with their signatures. These online wallets can be decrypted easily by the intruders/attackers which ultimately compromises the security of the coins. [8]
- **Offline wallet**: Offline wallets are also known as hardware wallets and have the same model as that of a regular wallet. If a regular wallet is stolen or gets misplaced then the chances of retrieving its contents are minimal or zero, likewise if an offline wallet is lost or stolen the bitcoins cannot be recovered [8].

- **Online plus offline wallet**: Just like an online wallet all the coins owned by bitcoin user are saved online the difference is that the signature used for transactions is not saved. Rather the signature is saved on another device which is not connected to a network, i.e. the signature is offline. The sole risk of using such

a wallet is that if the hardware in which the offline signature is saved gets misplaced or stolen, the security of the bitcoins is eventually compromised.

To avoid wallet theft, the recent wallets store the data or coins in an encrypted form contrary to the old wallets that stored bitcoins without any encryption. So, the newer wallets introduced to provide more security and safety. A wallet which fills all the above-mentioned loopholes and provides complete security and safety is yet to be created [8].

2.4.2 Anonymous or ownership theft

Although all the addresses are usually a set of random numbers and these numbers can't identify a person's identity, anonymity can still be compromised. For example, someone knows the address of a bitcoin user, then the address of the next bitcoin user can be guessed. Thus the possibility of identifying the owners arise and this can lead to tracing the history of a coin. In the end, the identity is disclosed.

2.4.3 Double spending attack

Double-spending is the result of successfully spending a bitcoin more than once. A set of bitcoins is claimed twice, once by the owner and second by the intruder. The intruder uses the bitcoin before the miner verifies the transaction for the owner. All of this happens within a short span of ten minutes. The double spending attacks work on the same model as that of a credit card fraud. This simply means that the intruder isolates the owner from the host network and makes him vulnerable to double spending. Double spending attacks are hard to achieve on the bitcoin network because of the computational

power requirements. Rather, it is easier to be a miner and earn the same amount of money than being an attacker [2].

2.4.4 Packet sniffing attack

In this type of attack an attacker monitors the transactions of a potential victim for a future theft. The attacker keeps track of the victim's network and at times can even view the transaction which originates from a particular node. This allows the attacker to attack the origin and can even lead him to more several other severe attacks.

2.4.5 Denial of service attack

These are the most common attacks on bitcoins. The attacks jam the network entirely which leads to the temporary shutdown of the system for a few hours. This ultimately results in the disruption of the service. This attack is the toughest to prevent. None of the companies have provided a complete solution to protect the system from denial of service attacks yet.

2.5 Overview of denial of service attack

This attack is an effort to shut down a network for a while. It can even be on a particular node which might be connected to the other nodes in a network. It floods the other nodes in a network. The flooding sometimes even suspend the services of the host connected to the internet. In case a server receives several requests from a single source and finds it suspicious or inclined towards an attack, the server automatically blocks that source. In order to ensure an attack is successful, attackers use several different sources to attack the

network so that the server remains unsuspicious of any such malicious activity and accepts requests of all transactions, thus leading to a successful attack. This type of attack which comes from more than one source is termed as distributed denial of service attacks. The distributed denial of service attacks over bitcoins have resulted in more loss than any other general attacks. The reason behind is that these attacks not only result in the theft of bitcoins but also in the loss of bitcoins due to the disruption in the network.

In the past years, there have been a lot of attacks on bitcoins. Bitcoins were introduced in the year 2011 and faced numerous attacks in the same year. We will be classifying the denial of service attacks on bitcoins since they occur at different levels and in different ways.

2.5.1 Classification of denial of service attacks over bitcoins

1. **Forwarding all the information to the adjacent nodes**: Bitcoin network works on TCP connections [9]. Since all nodes in the bitcoin system have to keep track of the incoming and outgoing connections, chances of the denial of service attack increases. So, to avoid them the node should send an edited list to the node adjacent to it. Since the edited list has limited information, if an attacker tracks this node, it will not be possible for the attacker to trace the connection back, and this precaution will eventually lead to the mitigation of denial of service attack or else the attack will easily be launched.

   The other denial of service mitigation technique used in this network is the implementation of the "**reputation based protocol in bitcoins**". These protocols

impose a penalty score for each connection and thus any node sending an invalid message is given a penalty score which keeps on increasing as the misbehavior of the connection increases and later resulting in the ban of that particular IP address for a day [9].

Alex Biryukov et al [9], in their paper discussed about the deanonymization and how the clients from the same IP address can be recognized. Their method unveils the anonymity of the clients even if they use any anonymity services like Tor and the method required for such an attack is simply related to logging the incoming traffic. All that an attacker requires is a few gigabytes of space and 50 connections to the server. The criticality level of this attack is high because it deanonymizes a bitcoin client's identity which is equivalent to compromising bitcoins attribute [9].

2. **Isolating a Bitcoin node**: This attack is done on a particular node. A single node in a bitcoin network has 117 incoming connections and only 8 outgoing connections. Although, not every node has both the connections, few have either of the connections or both. Since bitcoin network is an open, decentralized and independent of public key cryptography, it clearly depicts that no node is cryptographically identified. Therefore, all nodes are identified by their IP addresses [10]. And all the nodes having public IP's can accept 117 incoming connections and can have 8 outgoing connections as well.

This robustness and flexibility of nodes of accepting of many unique IP connections can sometimes lead to the joining of false nodes and this can lead to an attack on this peer to peer network.

Ethan Heilman et al [10], have come up with the name ‘eclipse attack’ in which the attacker controls or attacks on a particular node, floods it with a number of incoming connections and blocks the outgoing connections. Thus, the nodes get isolated from the network, without having an access to either the outgoing channel or any incoming channel. Since the adjacent nodes are unaware of this node’s isolation, the chain of passing the transaction information remains unhindered. This ultimately gives an attacker access to the block chain. The victim unknowingly wastes all of his energy or power. In this way, an attacker will be able to control the outgoing connections of the victim and connect it with the already acquired nodes, which will lead to all the 117 incoming connections being controlled by the attacker. The only loophole in this attack is that it will happen only when the nodes will have a public IP.

3. **Size of mining pools:** In a survey by Marie Vasek et al [11], came up with this type of distributed denial of service attack on bitcoins. They came up with this point that the denial of service attack even depends on mining pools (mining pools consist of transaction blocks). They stated that the pools which are bigger in size are more prone to distributed denial of service attacks than the pools which are comparatively smaller. To be very exact, the attacks which have suffered to bigger pools are 63% as compared to 17% of smaller pools. An example of bigger pool is Mt. Gox exchange. They concluded that usually all the attacks were done to stop the service flow in order to prevent the use of bitcoins.

4. Types of mining pools: The attacks in the recent days have been dependent on mining pools like Ant pool, Bw.com, nicehash, CKpool, and ghash.io. These few were recently attacked by the attackers [6]. We also categorize them as slush's pool, eclipse MC, and Eligius etc. The attacks were analyzed by the attackers. The attackers would attack only after observing the market share and the value of the pools.

Types of DoS attacks	**Name of the attack**	**Victim/ Compromisation**
Forwarding all the information to all the nodes	Anonymity theft	Anonymity can be compromised.
Isolation of node through denial of service	Sybil attack, Eclipse attack	Isolates a particular node from the network.
Denial of service on mining pools	Varying size attack	Mining pools bigger in size are attacked more often.
Denial of service on fluctuation of pools	Attacks on the basis of market share	Pools getting degraded or the attackers attack the pool whose market value gets fluctuated.

Table 2. 2 Classification of the denial of service attacks

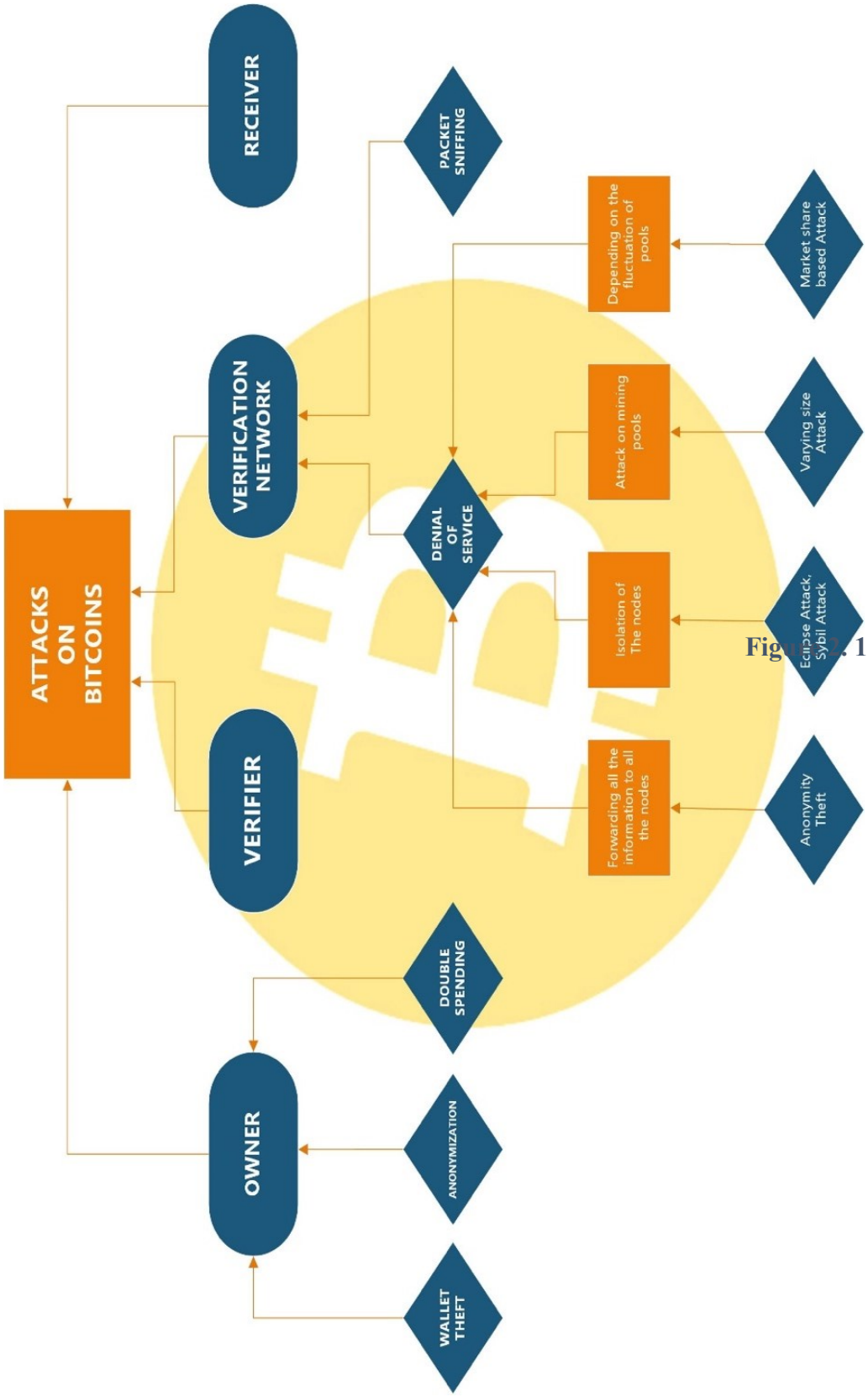

Figure 3. 1 Flowchart of attack

2.6 Existing authentication wallets and their limitations

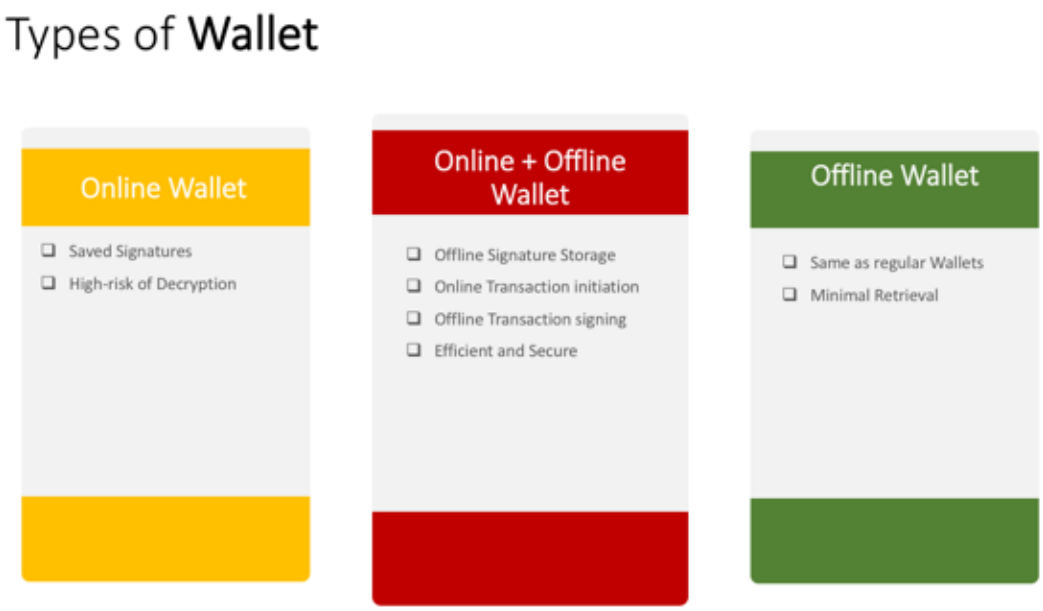

Figure 2. 2 Types of Wallets

There are various types of wallets available online and offline. We have different levels and authentication protocols that work on wallets. These wallets are based and developed according to the interface. Interfaces can be mobile, desktop, hardware and web-based. Keeping in mind and also considering the bitcoin wallets, the classification is as follows:

2.6.1 Mobile as an interface

All the wallets that are being used by mobiles are application based. In order to be used, those applications have to be installed on the device. Few of the wallets are pre installed in the user's phone or device which supports smart phone applications. Few wallets are pre installed on the device, such as Apple's passbook. Some of the applications

which can be installed are Isis, chirpify, lemon wallet etc. These wallets are used to store the users credit and debit card details. For using the wallets tap and pay system, we have to use the near field communication enabled smart devices. For all the other transaction which are online, we can either use the users card details and also for stores use, the application creates a barcode for the cards which can be scanned by the merchants, to use it for a transaction [12].

Trying to avoid to carry all the cards in a wallet and carrying a phone becomes a bit tricky and people tend to forget things sometimes. The above listed wallets are used in order to carry one accessory and interface for payments. In this way the problem of carrying multiple cards and physical currency is solved which eventually lead to safety from being robbed.

These are the wallets storing the details of our physical currency, now what about the wallets that are being used for cryptocurrency. Cryptocurrency such as a bitcoin which is not physically present have their own specific kind of wallets designed for them. They have their own features such as Centralized validation, decentralized validation simplified validation [13].

The privacy features are not the same in all the wallets. The attributes of a particular wallet differ from one another. The mobile wallets in bitcoins are of the following type:

1.GreenBits: This wallet uses two factor authentication scheme. It requires a little trust in the third parties for its payments i.e. simplified validation. The privacy to

which this wallet adheres is that it prevents the spying on the payments by changing or rotating the addresses during a transaction [13].

2.GreenAddress: This wallet has the same privacy as discussed in GreenBits, but with a slight change on the validation. It uses a decentralized validation technique which means, it connects with a random server from a list and verifies the payments but it is not as secure as Bitcoin core.

3.Coinomi Universal Wallet: Same privacy level as of GreenBits but it relies on centralized validation. Therefore, it requires a complete trust in a third party for verifying payments. Participation of a third party in a transaction makes this wallet vulnerable to attacks.

4.Bither: This wallet follows the simplified validation but the privacy level is weak which makes the wallet vulnerable to attack like packet sniffing, because of its incapability to prevent spying on its payments.

Similarly, there are many other wallets which follow either of the above wallets attributes (simplified, centralized, decentralized) to themselves as well. They can be named as Bitcoin wallet, copay etc.

The following table will give an overview of few of the mobile application wallet. It consists of the list of the wallets along with their functionalities and the way they are used.

Wallet name	Functioning
Apple passbook	Stores cards information and relies on scanning 2d barcodes.
Google Wallet	Stores card information and uses near field communication for tap and pay.
Lemon Wallet	Stores cards information, generates barcodes. Application is protected through a pin, to use.
Isis Wallet	Stores information of cards and is pin protected. It can be remotely stopped if stolen.
Chirpify	It uses PayPal to send and receive money. It requires signup on their website, in order to use the application.
PayPal	It stores cards information and is one of the most trusted and used wallets.

Table 2. 3 List of few of the mobile wallets

2.6.2 Hardware as an interface

These wallets are one of the most secure wallets. These wallets give the user full control over their money with an assurance of security. They are referred to offline wallets. Hardware wallets are as good as having a normal physical wallet. If the device is lost, money retrieval would be the minimum. The hardware wallets usually do not allow any software to be installed on their system which gives them an extra security against malwares and thus protecting them with attacks. Few of the wallets that can be named here are: Trezor, Ledger Nano, KeepKey [13].

2.6.3 Web as an interface

The web is one of the most vulnerable interfaces. It comes under the category of online wallets. The drawback of using the web or online wallet is that it requires centralized validation i.e. involving and trusting a third party. None of the existing wallets provides more than the basic privacy. The basic privacy prevents the sniffing of the packets but does not stop the involvement of the third party. The variability lies in the amount of control over the money and the authentication schemes.

The following table (Table 2.4) gives an overview of wallets with their functionalities as well as comparison with the scheme proposed in this thesis. The most trusted online(web) wallet which is being used by a number of people is PayPal. PayPal is the wallet which verifies the merchant before taking them or adding them to their network [14]. The verification is not necessary but verification makes the merchant more trustworthy and reliable for trades. That is why the verification process is free of charge.

PayPal requires signup in order to use the wallet. It stores all the information of the user's credit and debit cards of different banks or financial services. The storage of all the cards is on the basis of trust. The users have to trust this third party for all the transaction which happens in future. PayPal becomes a trusted authority for the user's online transaction that happens over a network i.e. worldwide.

Wallet name	Centralised validation	Two-factor authentication	Shared control over money	Money controlled by third party	Biometric used for authentication
Green Address	✓	✓	✓		
BitGo	✓	✓	✓		
Coinapult	✓			✓	
Circle	✓	✓		✓	
Xapo	✓			✓	
Coinkite	✓			✓	
PayPal	✓		✓		
ApplePassbook	✓		✓		
Google Wallet	✓		✓		
SWAP	✓	✓	✓		✓

Table 2. 4 List of few of the wallets with their functionalities

2.7 Summary and motivation

As we came across in the papers that bitcoin is a very popular crypto currency. All the papers give valuable insights into the functioning of a bitcoin network along with the general attacks associated with it. Bitcoin is decentralized digital currency which allows a peer to peer online transaction independent of any third party i.e. financial institution. The use of bitcoin assures secure transfer and easy access to online transactions. Although bitcoin is a relatively new digital currency, it has been accepted by a growing number of institutions and users throughout the world. One such well-known institution is the Wikipedia foundation that accepts donation through bitcoins [15]. Although bitcoin offers a rational amount of security towards users. The architecture is still in its early stages of development.

Past incidences have identified technical challenges as well as legal risks of using bitcoins. Any bitcoin user should have a clear understanding of how to legally manage transactions and should be aware of the security risks associated with it. The thesis further throws light on the working of a wallet, threats related to it and all the security issues associated with it. The threats have been categorized with respect to the different phases of the transaction as well as the stakeholders in a bitcoin transaction. We have tried to cover a generalized overview of attacks with respect to the Owner, receiver, verifiers, and verification network in the literature survey. Further classifying the subdivision of attacks with respect to the classes of a particular leg of it. Some of the

loopholes in the security standards of the bitcoin network have been discussed in this thesis. As a young project, some of the modules of bitcoins are constantly improving.

With the motive of removing the flaw which is currently going on in the bitcoin wallet technology, the proposed scheme can also be targeted for the general digital wallets which are dealing with the storage of cards of the users. This technology can further be implemented in the current wallet such as PayPal for the two step verification as well as for the authentication scheme. This will remove the security threat to a certain extent.

Thus, with the use of proposed scheme, almost all the wallets following the two step authentication scheme can be protected and given an extra layer of security to avoid the client side thefts eventually leading to use of more client side based digital wallets.

Chapter 3: System Design

3.1 Need for Biometric authentication

Authentication is the term used for assuring or confirming that a communication is authentic. It is categorized into two types:

3.1.1 Peer authentication

This authentication is provided at the time of connection establishment and data transfer. In general, when both the peers are in communication with each other and have the same protocols implemented on both of the system [16].

3.1.2 Data origin authentication

This authentication provides the authenticity of the data source. The source is the phase from where the data is coming or originating. In this type of authentication, the peers need not be in communication. For example: electronic mails [16].

Authentication which is often referred as authenticating or validating a user identity [17]. In order to verify and validate a user, using passwords as the only way to authenticate is not feasible anymore. Day by day with the advancement in technology, the introduction of several new tutorials and also the websites etc, have made users create a separate login id's and passwords. Most of the users create a common password to almost all of their usernames. This leads them to be a victim of social engineering, phishing and to a number of other attacks.

In order to protect them from these type of attacks, using the same passwords, two-step verification was introduced. In this method, the intruders or attackers cannot attack user, even if they have their passwords. How?

In a two-step verification, the user has to sign-in in two steps. The first step is the password, which leads to the second step. The second step depends on the user's choice. Either a one-time password can be sent to the user via text or via call or a security, a key could be provided. The security key is saved in the user's thumb drive via a universal serial bus port on his computer [18].

Once the user is secured via the two-step verification, even if the password is hacked, or has been leaked through shoulder surfing, the user remains protected. Now the question arises, what if the device through which the user signed up for the two-step verification is lost? There is always a possibility of losing the device and the password at the same time. What if the attacker is from the user's hostel, or within the same organization, or even from his near by area or close surroundings. This is where the need of a biometric authentication arises.

Using a biometric authentication in the signing-in process will add an extra layer of security. And this will be security with an ease to use. Imagine, neither needing to remember passwords nor answering security questions. Time saving is an added advantage. The user can sign in with just authenticating himself with his body parts like finger, iris etc. The password is always with the user or within him. This involves no risk

of losing it or forgetting it. This makes biometric authentication an added layer of security making the technology convenient to use and faster to compute.

3.2 Working of SWAP

"Secure wallet authentication protocol" is the name of the protocol proposed in this thesis. This protocol is a novel authentication algorithm for users in a wallet scheme. The algorithm focusses on the input and the signature sent and received during and for initiating a transaction. Our algorithm is focussing on the client-side wallet scheme. The communication between a wallet and a server.

3.2.1 Root of the idea

The offline wallets are considered to be a secure way of signing and doing a transaction i.e. storing the signature in an offline device and performing the transaction in an online device and signing it off from an offline device where the signature is stored. This method is a secure way of signing a transaction, as long as the security of the device is not compromised. The moment, the security is compromised, this method becomes an easy way to perform the transactions by the attackers. To the best of our knowledge, this is the best possible solution to the problem addressed above, the authentication scheme proposed in this thesis i.e. SWAP.

3.2.2 Algorithm

In SWAP, we are using the biometric identification of users to initiate and proceed with the transactions. This way we are making it more secure than the earlier wallets. We have taken the idea of recording the signature in an offline device and not to store it that device. In the proposed approach, we have taken wallet server communication in two phases i.e. wallet being the client here, and assuming that the client-server communication is secure, the first phase is the registration, and following is the transaction phase.

1. **Registration phase**: In this phase, we are taking in account our first assumption i.e. secure client-server communication. Wallet and server will have a public key cryptosystem in order to have a secure communication between them. The initial step would be from the wallet side. The wallet will acquire the biometric identification from the user and will take that to convert the hash of it. After converting it to a hash, the wallet will take the hash of the biometric identification and will pass it through a random number generator making the probability of being attacked less, thus increasing the degree of randomness. We will then compute a signature SignW which would be the keyed hash of the hash of the biometric identification. The seed over here would be the public key of the wallet. Now, the first message will be sent consisting of the encrypted hash of the biometric identification and the encrypted keyed hash i.e. SignW and they will be encrypted with the public key of server and private key of server and private key of wallet respectively.

Now, the server side computation will include the verification of the wallet. The server will have the keys on its side. The initial step on the server side would include the decryption of the hash of the biometric identification. After decryption, the server will compute the signature just the same way as performed by the wallet and match the signature with SignW.

If the signature matches, this will lead to the step of random number generation. The server will have the pseudo random generator function which would be same as a wallet, thus leading to the formation of the same random number. The server will send a signature SignS which would be the keyed hash of random number and the key being the public key of the server. The server sends the acknowledgement along with the signature.

When the message is received by the wallet, it verifies the wallet by computing the signature with the random number and comparing both the signature i.e. SignS and the signature formed. After this step, both the wallet (client) and server are synchronised. Now, the device will delete hash of the biometric identification with the biometric identification.

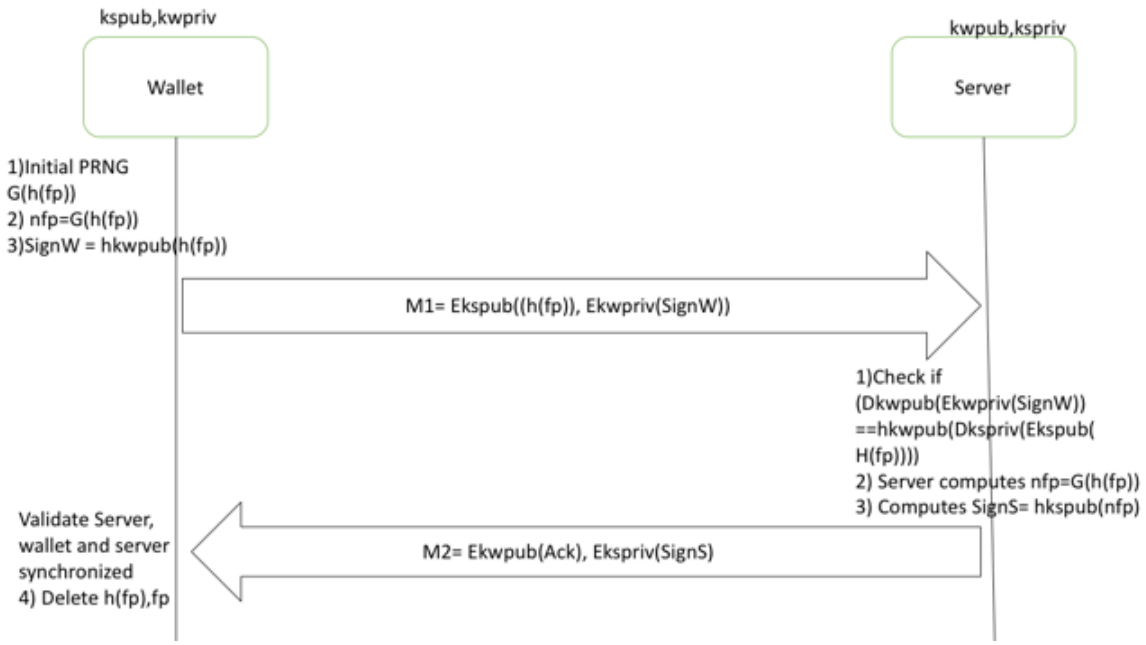

Figure 3. 1 Registration Phase

2. **Transaction phase**: In this phase, when the transaction is initiated. The server will send the encrypted hash of the biometric identification (BI) and the wallet will acquire the BI and convert it to the hash of it. The protocol generates the random number of the hash of the BI. In the next step, the random number and the hash of the BI are XORed. After XORing the protocol will compute the signature of the transaction i.e. SignW Txn which would be the keyed hash of the hash of the XORed function. Now, what so ever is the document that signs the transaction (be it signature or a document) which is to be sent to the server side for signing the transaction is sent by encrypting the document with the transaction key i.e. random number generated by the hash of the BI.

   Now, the server will verify the wallet and validate the transaction by computing the signature in the same way as performed on the wallet side. The same

mathematical steps are followed and then the signature formed is compared with the SignW Txn. If it matches, the wallet is verified. The second step is to make the random number as the transaction key for decrypting the transaction document and validating the transaction.

After the verification and validation, the server will update the random number variable as hash of the XORed operations of hash of BI and random number. The same steps are also followed on the wallet side as soon as the acknowledgement is received by the wallet. Thus, this will lead to the last step of the protocol i.e. to delete the hash of the BI, BI, and the transaction key. Which will lead to the security of the user even if the device's security is compromised.

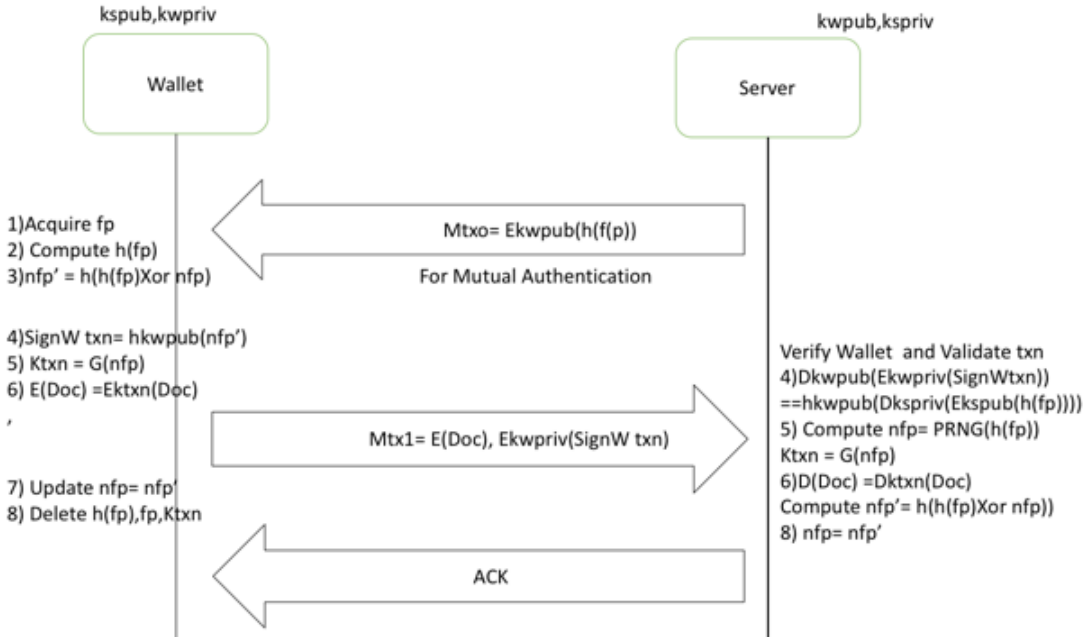

Figure 3. 2 Transaction phase

3.3 Uniqueness in SWAP (secure wallet authentication protocol)

With all the wallets, we have seen in section 2.4. None of them were providing security of the hardware. In the above section, we have discussed the different wallets and their attributes with reference to the authentication, validation, and privacy. The online wallets were vulnerable to a number of attacks launched online, making them lie in the category of risk in use. The hardware wallets are as good as normal physical wallets which are used to store physical currency. If the device is lost, the user loses all the money and this sometimes even lead them to lose their digital signatures. Thus, the use of hardware wallets makes it secure only till the device security is uncompromised. The moment hardware is compromised; the security is breached. Wallets having two-step authentication system are also not completely secure. If the device which is registered for the two-step authentication, whatsoever it may be, if that device is compromised, the security is breached. Thus, these attacks will lead you to be a victim of digital wallets.

To avoid this threat from users, we have proposed a new authentication algorithm which is running the wallet on a different scheme. Our scheme involves the use of biometric, which makes them safer from many threats. One of which would be, if the users lose their wallets or hardware or device, it is not going to trigger the device to function. It will run only with the legit user's biometric. Involving our own authentication protocol for the authentication scheme makes it unique from others.

Using the concept of hash chaining, dynamic key generation and deleting the biometric data from the device makes it difficult for the attacker to attack the wallet. Deleting the values from the device after use makes it unique and more secure. The use of hash chaining for the transactions makes the attackers confuse about what's going in the transaction. The attacker cannot go either ways, neither ahead in the chain or back in the chain. The protocol's feature of working on the output of the previous transaction and using it for the next transaction using it together with the previous value of the transaction.

The attacker unaware of the previous data because the data is hashed. The combination of hash chaining with the dynamic key generation technique makes the algorithm secure. The combination makes the attackers probability to succeed or breach the system to minimal. Thus, these are the features making it more feasible to apply in the wallet system.

3.4 Limitations in the protocol (Hash Collision)

A hash collision occurs when two input values produce the same hash or cryptographic digest. The probability of having a hash collision in SHA 256 has been significantly reduced. The number of attacks that have happened in the last ten years on SHA 1 is none. The last reported collision was in 2005. SHA 2 is similar to SHA 1 but the attacks that have happened on SHA1 have not been extended to SHA 2. The probability of hash collision in our implementation is not possible, considering the statistics. Considering the other side, the sensor might take us to hash collision due to its image capturing

technology. This flaw or limitation can be removed with the use of a better sensor which can capture or record the fingerprint data accurately [19].

Chapter 4: Implementation

We have implemented the proof of concept (POC) using the Arduino board, fingerprint sensors as the BI. Connecting both of them serially with our laptop. Implementing the client server architecture on the same laptop.

We have used Arduino to help the fingerprint sensor communicate with the laptop. We had two other options to implement the POC. Raspberry pi and FTDI breakout being the other two ways to make the fingerprint sensors communicate with the laptop serially.

4.1 Arduino

The first reason to choose Arduino is the fact that it is considered to be one of the most feasible interaction prototypes for innovations and experimentations. Arduino is an open source prototyping model which is used in the implementation of the projects using hardware and software. These boards have the ability to receive, transmit and process signals. This microcontroller can be used by people not having electronics background as well. Since the board can be communicated via an IDE i.e. Arduino software makes it easier to use. The simple to use attribute of the board makes it popular than any other board or microcontrollers in market. It can be used on a Mac, Windows and Linux [20]. Another favourable reason for a user to buy it is its low-cost feature.

In this POC, I have used UNO Arduino microcontroller. It is very robust and is based on ATmega328P. It has 6 analog inputs, 14 digital input/output pins, a USB port, 16 MHz

quartz crystal and a reset button. It can be powered up by either connecting it with a USB or the AC to DC power adaptor [21]. It can be used with the Arduino IDE. The UNO Arduino had a few drawbacks, it is not integrated with a Wi-Fi and memory card slot which makes its usability in a shorter range. It has a ram of 2 kb and flash memory of 32kb. The operating voltage is 5V [21], it also has an option of 3.3 V.

Figure 4. 1 Arduino UNO

4.2 Fingerprint scanner

The fingerprint sensors can be really tricky to choose and variations in the capturing technique make it even more difficult. The fingerprint sensor from Sparkfun is selected for the POC. Sparkfun TTL GT511C [22] is a reliable and has a 32 bit CPU inside the scanner, enabling it to compute effectively and efficiently. The fingerprint scanner works on optical sensors i.e. it captures the image of the finger. At the time of registration, the scanner asks the user to scan his finger thrice, in order to capture the impression from different angles. This increases the accuracy of the sensors to detect an identification.

The fingerprint scanner works on the principle of image integration. The device takes the image of the finger three times, integrating them together into a single image and making it a template. Now, when the user has to be verified, it matches the fingerprint with the template. If only, the fingerprint is a subset of the template, the scanner detects and the user is verified. The beauty of the product also lies in the fact that it captures the image almost equal to the size of 20 kb, but at the time of creating a template, it compresses it to a size of 500 bytes. This device's memory can store up to 200 fingerprints.

Figure 4. 2 Fingerprint Scanner

4.3 Limitations of the hardware devices

- Arduino Uno has a memory restriction of 32 kb which requires the program to be uploaded repeatedly in order to function.
- When it comes to detecting a fingerprint, the fingerprint sensor has a limitation: the sensors cannot calculate the accuracy of a finger with more grooves or fingers with dirt/sweat on them.

Now, coming to the software and programming languages used in this POC. We have used the Arduino IDE, downloaded from their website and used it with the fingerprint sensor. We also took assistance of libraries with which the fingerprint sensor executes the functionalities. The coding for registration and transaction is done in python. We have used the name of the ide to write the python codes. The steps have been followed in the same order as referred in the algorithm section. For implementing the public key cryptosystem, we have used the RSA cryptosystems, the reason being the practicality of this cryptosystem. RSA stands for Ron Rivest, Adi Shamir, Leonard Adleman. To create the hashes, we have used the SHA 256 (secure hash algorithm), due to its use in highly secured applications like TLS and SSL. SHA 256 produces a digest size of 256 bits.

Following down the line, the symmetric key encryption used in the algorithm is the AES (advanced encryption standard). It is based on the Rijndael cipher. It is widely used in security applications by the US government for passing the top secret information and also used by the national security agencies [23]. The above hash functions and encryption standard is followed and is implemented in the algorithm, making it secure from a number of attacks.

4.4 Security of the scheme

With the intriguing combination of the BI, Hash chaining and the dynamic key generation. The scheme addresses a number of problems being faced by a number of wallets in the current market, be it online or offline.

The use of BI makes it easier to use. The conversion of the real BI value to the hash of it and producing a pseudorandom number by taking it as a seed, makes it secure. The signature produced on the wallet and server sides are produced dynamically with the BI value which increases the randomness to the next level. When the transaction phase is initiated, the wallet creates a dynamic key which keeps on updating with every transaction.

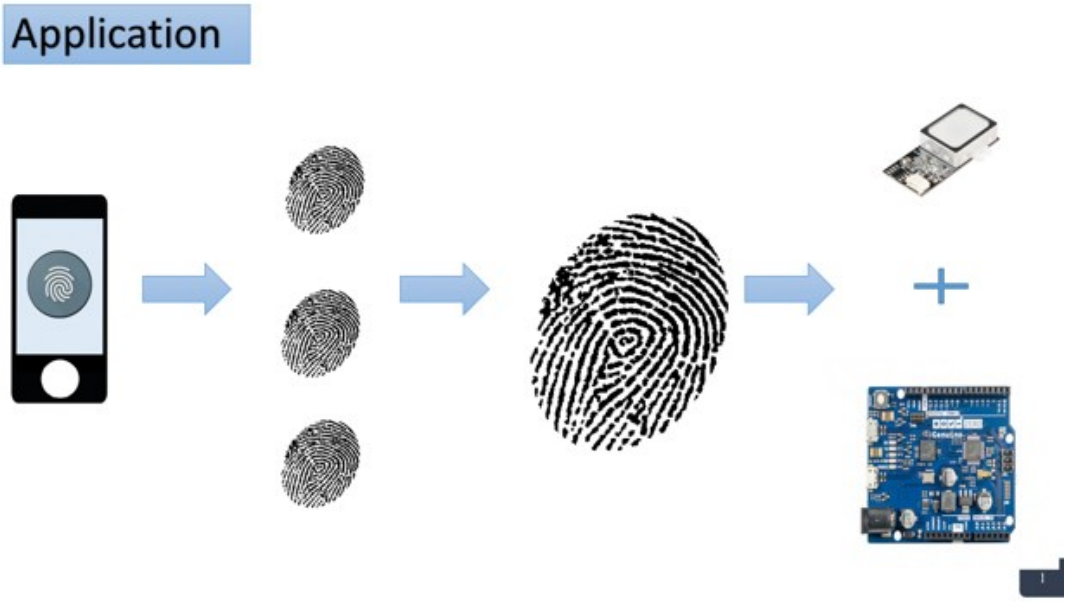

Figure 4. 3 Functioning of the fingerprint sensor and the Arduino

The value used for the key is the same random number generated with the hash of the BI as the seed. This key value is changed after every transaction as the other parameters are affected. The random number generated is updated after every transaction which is the hash of combined XORed of a hash of the BI and the random generated in that phase before that particular transaction in that cycle. This value is updated on both the sides i.e. wallet and server. This brings in the concept of hash chaining which helps in mitigating man in the middle attack.

In the end, the wallet deletes the BI, transaction key and also the hash of the BI. This increases the wallet level security. In the case of a wallet theft i.e. physically, still a transaction would not be possible to initiate. This also helps to secure the BI values of a particular user. Which eventually leads to the prevention of loss of monetary funds. Thus, to the best of our knowledge, the scheme proposed here secures the wallets in a better way than most of the wallets currently being used.

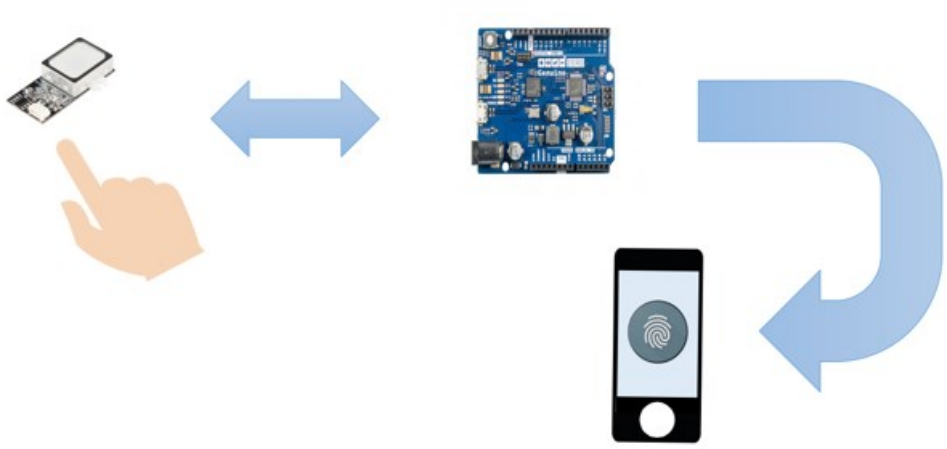

Figure 4. 4 Communication between the devices

Chapter 5: Evaluation and Results

In order to evaluate this scheme, we have used the protocol verification tool scyther [24]. We also evaluated our protocol with respect to the security goals [16] for verifying the scheme, namely confidentiality and integrity. We also tested for 'man in the middle attacks'. With reference to integrity, the scyther tool verifies the types of attacks during communication, which makes sure that the signature sent by the wallet or the servers are received exactly in the same format at each end.

Scyther is a "black box" testing protocol suite. Scyther is a protocol tool for formal analysis of security protocol [24]. It tries to perform most of the attacks that can happen on a communication channel. Such as, replay attack, man in the middle attack which can be session hijacking, modification and etc. It follows certain adversary characteristics. It follows the Dolev Yao model. This model assumes that the cryptography is perfect, messages are the abstract terms and that the attacker has full control over the communication network.

Following the formal protocol analysis using scyther, we will also discuss the performance of our protocol with respect to other security goals such as non-repudiation, access control and forward-backward secrecy. The only way to initiate a transaction is the use of the BI impression of the user, and if the user does not agree or permits, the transaction is not sanctioned. The unique attribute of the scheme to compute the signatures dynamically with the use of BI of the user satisfies the goal of

non-repudiation. By mutually authenticating the user with the server before starting a transaction deploys the goal of access control. Checking the user's information by the BI at the entry level of wallet with the stored data on the server prevents the attacker from breaching the security.

The use of hash chaining in the transaction phase is implementing an important security feature which is forward and backward secrecy. The step of computing hash with SHA256 makes it irreversible. The combined hash of hash of BI and the random number implements forward and backward secrecy. Even if the attackers gets the information about a particular session, they will neither be able to go forward nor backward in a transaction. This is how hash chaining acts in the security.

While using the scyther tool, we had to keep in mind about the phases while testing out the protocol. The testing is conducted in two phases. For the registration phase, we have changed the parameters. Since the registration process goes only for one time. That's why the maximum number of runs is set to one. In the advanced parameter, the search pruning is set to find all attacks. The setting is displayed in the table below:

Maximum number of runs	1
Match type	Typed matching
Search pruning	Find all the attacks
Maximum number of patterns per claim	10

Table 5. 1 Parameter settings for Registration Phase

After setting the parameters, we verified the protocols and the result showed no attacks within bounds as shown in the screen shot below:

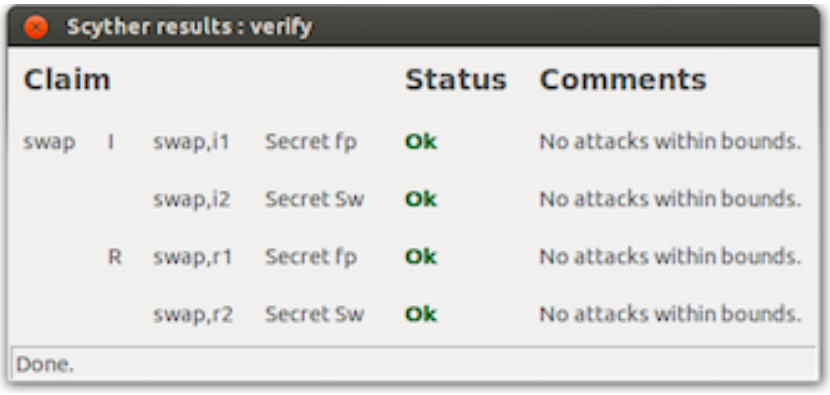

Figure 5. 1 Result for Registration Phase

We follow the same procedure for the transaction phase, but we change the parameter here again. Since, the transaction phase has to occur more than once, so we change the number of runs as 10 and rest of the parameters remain the same. Thus, this result also shows no attacks and the screen shot shown below reflects the exact result.

Maximum number of runs	10
Match type	Typed matching
Search pruning	Find all the attacks
Maximum number of patterns per claim	10

Table 5. 2 Parameter Setting for Transaction Phase

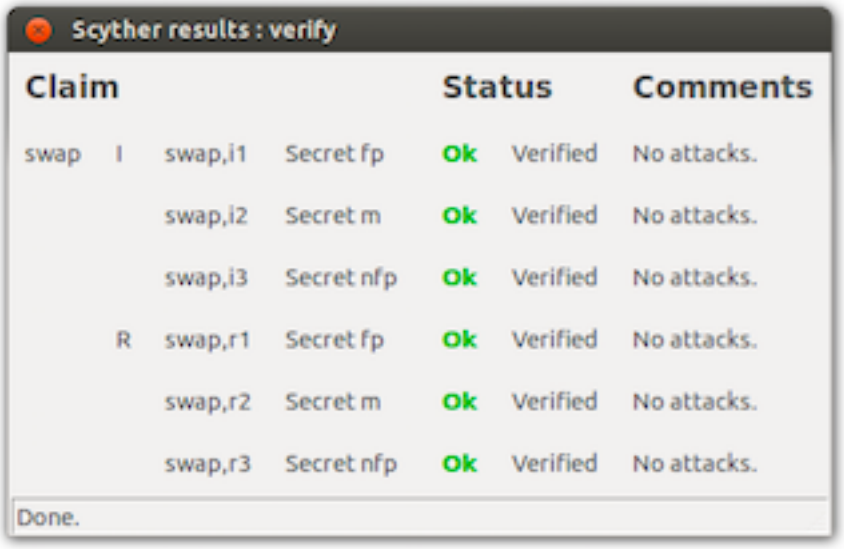

Figure 5. 2 Result for Transaction Phase

The result reflects the claims made in the communication protocol are true. There is no attack that can be launched.

In this chapter, we have discussed the implementation of the security goals. We have analyzed our scheme through the security features and tested through the communication protocol analyzer tool i.e. scyther.

Chapter 6: Conclusions and Future Work

6.1 Conclusions

In our research, we have proposed a novel and secure authentication protocol for wallets. Our algorithm makes use of BI to identify the user and initiates the transaction without the use of any additional security measures. It follows the tag line of “security with ease”. The scheme is implemented through Arduino and sensor data from the BI scanner.

The POC is implemented on Arduino, with the help of fingerprint sensor. At the time of transaction or registration, the POC has to be in connection with the internet. For the first time, the device when connected to the internet, takes the BI of the user to register it with the server. Once the registration is successful, the users have to scan their finger for each and every transaction. The POC is a stand alone device which is not connected to the internet.

In order to evaluate our scheme, we have used the protocol analyzer tool (Scyther) and successfully passed all the claims. The protocol has no attacks within the bounds for registration phase and no attacks for the transaction phase during communication. The results depict that the claims made during the communication are all valid and verified. Thus, the proposed scheme satisfies the security goals and implements security with ease.

Our protocol is running on the assumption that the server is secure. The registration phase is executing once for a particular user.

6.2 Future Work

The authentication scheme is proposed for wallets. In the future, it can be implemented in numerous other fields. The scheme can be embedded in mobile devices and can be integrated with the applications. It has a wide scope to be implemented on ATMs. ATMs can be configured with biometric sensors, and those sensors can be used to verify the identity of a user. The scheme is not only limited to banking or monetary transactions. It can also be used to transfer files or data by using a standalone device with an appropriate memory size.

Figure 6. 1 Future Work (Application)

Bibliography

[1] The New York Times, " A shift towards digital currency," [Online]. Available: http://www.nytimes.com/roomfordebate/2012/04/04/bringing-dollars-and-cents-into-this-century/a-shift-toward-digital-currency. [Accessed 21 March 2016].

[2] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Accessed 14 July 2015].

[3] Coindesk, "How bitcoin mining works," [Online]. Available: http://www.coindesk.com/information/how-bitcoin-mining-works/ . [Accessed 19 July 2015].

[4] Coindesk, "How do bitcoin transactions work," [Online]. Available: http://www.coindesk.com/information/how-do-bitcoin-transactions-work/. [Accessed 23 July 2015].

[5] Coindesk, "What is bitcoin," [Online]. Available: http://www.coindesk.com/information/what-is-bitcoin/ . [Accessed 13 July 2015].

[6] S. Higgins, "Bitcoin Mining Pools Targeted in Wave of DDOS Attacks," [Online]. Available: http://www.coindesk.com/bitcoin-mining-pools-ddos-attacks/. [Accessed 16 July 2015].

[7] Bitcoin forum, "List of bitcoin heists," [Online]. Available: https://bitcointalk.org/index.php?topic=83794.. [Accessed 28 July 2015].

[8] Bitcoin.org, "Secure your wallet," [Online]. Available: https://bitcoin.org/en/secure-your-wallet. [Accessed 28 September 2015].

[9] D. K. I. P. Alex Biryukov, "Deanonymisation of clients in Bitcoin P2P network," in *CCS '14: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014.

[10] A. K. A. Z. S. G. Ethan Heilman, "Eclipse Attacks on Bitcoin's Peer-to-Peer Network," in *24th USENIX Security Symposium*, Washington, 2015.

[11] M. T. T. M. Marie Vasek, "Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem," in *Financial Cryptography and Data Security*, 2014, pp. 57-71.

[12] Hongkiat, "Digital wallets: 10 mobile payment systems to take you There.," [Online]. Available: http://www.hongkiat.com/blog/digital-wallets/. [Accessed 2 March 2016].

[13] Bitcoin.org, "Choose your Wallet," [Online]. Available: https://bitcoin.org/en/choose-your-wallet. [Accessed 27 October 2015].

[14] Paypal, "The verification process," [Online]. Available: https://www.paypal.com/ca/webapps/mpp/security/verification-faq. [Accessed 27 January 2016].

[15] wikimedia, "Ways to give," [Online]. Available: https://wikimediafoundation.org/wiki/Ways_to_Give#bitcoin. [Accessed 28 January 2016].

[16] W. Stalling, Network Security principles and practice, 5th edition ed., Pearson, pp. 21-22.

[17] Taylor and Francis group, LLC, Mechanics of user Identification and Authentication, 2007, p. 4.

[18] Google, "2-step verification," [Online]. Available: https://www.google.ca/landing/2step/#tab=how-it-works. [Accessed 28 February 2016].

[19] www.wikipdia.com, "SHA-2," [Online]. Available: https://en.wikipedia.org/wiki/SHA-2. [Accessed 7 April 2016].

[20] Arduino, [Online]. Available: https://www.arduino.cc. [Accessed 24 December 2015].

[21] Arduino , "Arduino UNO," [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno. [Accessed 5 January 2016].

[22] Sparkfun, "Fingerprint scanner," [Online]. Available: https://www.sparkfun.com/products/11792. [Accessed 29 January 2016].

[23] wikipedia, "Advanced encryption standard," [Online]. Available: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard. [Accessed 31 January 2016].

[24] C. Cremers, "Department of Computer Science," University of Oxford, [Online]. Available: https://www.cs.ox.ac.uk/people/cas.cremers/scyther/. [Accessed 5 March 2016].