COMP4801 Intermediate Report
University of Hong Kong
Final Year Project 2018-19
14-04-2019
SECURE DIGITAL WALLET DESIGN

Name: Aditya Sethi
UID: 3035240063
Supervisor: Dr. J.T.H Yuen
Abstract

Over the last couple of years, investors have started trusting the digital modes of payment. The Government of India has taken a step forward to move towards a cashless economy. They have started to realize that digital modes of payment are fast and also secure at the same time. Investors have started to become bullish about cryptocurrencies. This optimism about the future of cryptocurrencies gave rise to crypto wallet service providers. A crypto wallet is an interface used by investors to buy and sell cryptocurrencies. With a lot of companies offering the service on platforms like web (web application) or mobile, there has been rising concerns about the security that these systems provide. Web or mobile wallets are connected to the internet and are thus prone to attacks. The lack of sophisticated security mechanisms in these systems prevent investors from accepting the technology.

This progress report discusses how crypto wallets came into being. It highlights the importance of having secure crypto wallets. It also explains the development process of a secure cryptocurrency wallet. This project explores various techniques or technologies that can be used to enhance the security of online crypto wallets. The report discusses the following features; Two-Factor Authentication, Hierarchical Deterministic Wallets and Multisignature technology. The wallet will incorporate these features to make the web wallet a secure platform for the investors.
Acknowledgment

I would like to express my sincere gratitude to my supervisor Dr. J.T.H Yuen for providing invaluable expertise and constant support in all my endeavors to make this project a success.

Lastly, I would also like to thank my friends and family for constant motivation.
# Table of contents

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>2</td>
</tr>
<tr>
<td>Acknowledgment</td>
<td>3</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>6</td>
</tr>
<tr>
<td>1.1. Background: The rise of cryptocurrencies and crypto wallets</td>
<td>6</td>
</tr>
<tr>
<td>1.2. Motivation</td>
<td>7</td>
</tr>
<tr>
<td>1.3. Objective</td>
<td>7</td>
</tr>
<tr>
<td>1.4. Significance of the project</td>
<td>8</td>
</tr>
<tr>
<td>1.5. Report Outline</td>
<td>8</td>
</tr>
<tr>
<td>2. Literature review</td>
<td>9</td>
</tr>
<tr>
<td>2.1. Blockchain</td>
<td>9</td>
</tr>
<tr>
<td>2.2. Public key cryptography</td>
<td>12</td>
</tr>
<tr>
<td>2.3. Cryptographic Hashing</td>
<td>12</td>
</tr>
<tr>
<td>2.4. How does hashing provide data integrity?</td>
<td>12</td>
</tr>
<tr>
<td>2.5. Digital Signatures</td>
<td>13</td>
</tr>
<tr>
<td>2.6. Signing and Verification: How digital signatures are verified</td>
<td>13</td>
</tr>
<tr>
<td>2.7. Cryptocurrency: An application of Blockchain</td>
<td>13</td>
</tr>
<tr>
<td>2.8. Wallets in the market</td>
<td>14</td>
</tr>
<tr>
<td>Web wallet</td>
<td>14</td>
</tr>
<tr>
<td>Mobile wallets</td>
<td>15</td>
</tr>
<tr>
<td>Desktop Wallets</td>
<td>15</td>
</tr>
<tr>
<td>Paper Wallets</td>
<td>15</td>
</tr>
<tr>
<td>Hardware wallets</td>
<td>15</td>
</tr>
<tr>
<td>3. Methodology</td>
<td>17</td>
</tr>
<tr>
<td>4. Features</td>
<td>19</td>
</tr>
<tr>
<td>5. Application functionalities</td>
<td>29</td>
</tr>
<tr>
<td>6. Difficulties Encountered</td>
<td>36</td>
</tr>
<tr>
<td>7. Project progress</td>
<td>37</td>
</tr>
<tr>
<td>8. Future Steps</td>
<td>39</td>
</tr>
<tr>
<td>9. Conclusion</td>
<td>40</td>
</tr>
<tr>
<td>10. References</td>
<td>41</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Blockchain consisting of blocks linked together 9
3.1 System Architecture 11
4.1 Barcode Authenticator 11
4.2. Barcode scan 12
4.3 6-digit pin generation 13
4.4 Multikey locker 13
4.5 HD wallet key generation process 23
1. Introduction

1.1. Background: The rise of cryptocurrencies and crypto wallets

The cryptocurrency phenomenon traces its roots back to 2009. An anonymous personality named Satoshi Nakamoto invented Bitcoin which he referred to as a “Peer-to-Peer Electronic Cash System”. The idea of developing an electronic cash system was not new. David Chaum, a computer scientist was the first one to introduce the concept of e-Cash back in 1982. He published a paper entitled “Blind signatures for untraceable payments” that illustrated a new form of cryptography that claimed to provide anonymity to the transactions. He wanted to develop a peer-to-peer payment system that would hide payment information from third parties. He leveraged his research and created the company DigiCash to create a secure online currency. The product was not widely accepted by the society but this was certainly a start of a new revolution in the digital payment industry. Chaum opened gates for other cyber analysts.

The year 1998 witnessed an emergence of some cryptocurrency ideas. Out of the many ideas, the one that attracted a lot of people was Nick Szabos’ idea of developing a decentralized digital currency in the year 1998. The main idea was to remove the intermediaries from the transaction chain. Although the idea was never implemented it is often referred to as the precursor to Satoshi Nakamoto’s Bitcoin Protocol.

A decade later came Bitcoin in the year 2008. It changed the way how people trusted digital currencies. The security that the underlying technology Blockchain provides grabbed everyone's attention. With this technology no one’s in charge because everyone’s in charge. If one entity tries to tamper with the system others get to know. Because of this trait the technology is referred to as the technology of consensus. Entities have to unanimously agree on the legitimacy of the data before it becomes a part of the public ledger. ledger to put simply, is a database of all the transactions available with every node in the Blockchain network.

Considering the level of security that the technology provides people started trusting it more. Increase in the investments gave rise to wallet service providers. A wallet service provider connects the investor to the bitcoin network. Cryptocurrency wallets are used to access, store, send or receive cryptocurrencies such as Bitcoin, Ethereum, Litecoin and many more which would not
be possible otherwise. Cryptocurrency wallet is a software program that stores private and public keys. It is a link to various Blockchain’s in the market and it allows the users to connect to these Blockchain’s and perform different actions like checking wallet balance, sending or receiving cryptocurrencies. Blockchain will be explained in detail in the literature review section.

1.2. Motivation

With the increasing number of the digital payment platforms in the market there have been rising concerns about the security protocols being used by these technologies. There have been numerous cases where people have lost enormous amounts of funds due the lack of proper security standards being followed. There are two main causes of such thefts; the first cause relates to user negligence and the second, the loopholes in the security systems of the application. Investor often lose their funds because of their own mistakes. When a user signs up for an account, he/she tends to click photos of the passwords or tends to send passwords by email. If they forget their passwords, they try to recreate it by getting an OTP on their mobile phone without realizing that even their phones can get hacked. This proves that one cannot just rely on traditional methods of using username and password to authenticate the user. There is a need to integrate better security measures. Weak authentication is a threat to the digital payment industry. Some security measures can be taken on the application level also to prevent the user from making such mistakes. Resolving the security concerns in the crypto payment industry is the motivation behind this project.

1.3. Objective

The goal of the project is to make a fully functional secure crypto wallet. The task of accomplishing this target will be divided into two parts. The first part will deal with the normal functioning of the wallet without taking into consideration the security part of the wallet. Basic features such as transferring cryptocurrencies from one account to another, broadcasting transactions to the network and checking wallet balance will be implemented first. The second goal of the project will be to integrate different security features like two-factor authentication, seed generation and multisignature to the wallet to make it secure.
1.4. Significance of the project

This project will aim to integrate new security features to the cryptocurrency wallet. This project will illustrate how people can trust the wallet service by understanding the enhanced security that it will provide to the users. This project will be a step towards making the digital wallet industry more acceptable. (even though people are doing transactions, still some are skeptical about its use, this is going to be the future of digital economy)

1.5. Report Outline

This report is divided into five chapters. Chapter one provides the background. It explains how digital currencies came into being and how the cryptocurrencies got into the forefront of the Fintech industry. Chapter one discusses the motivation behind the project and sets out the objective for the project.

Chapter two presents a literature review by explaining the key technologies and the concepts associated with developing a crypto wallet. It starts with explaining how the Blockchain technology works. It then explains the key terms that the reader should be familiar. A good understanding of these concepts provides basis for understanding how a cryptocurrency wallet works and how it can be developed. The chapter will also throw some light on what types of wallets exist in the market and what are the benefits and disadvantages of using them.

Chapter three explains how the wallet application will be developed. It starts with explaining the architecture of the wallet. It then explains what all technologies will be used to develop the front-end, back-end and the database.

Chapter four will explain in detail the security features that will be integrated to the wallet application. It will also provide a step by step explanation as to how the features will be developed.

Chapter five will display what the application offers to the users.

Chapter six discusses the difficulties encountered during development.

Chapter seven discusses the project progress. Chapter 8 discusses the future steps.

Chapter nine concludes the report by giving an overview of the whole project.
2. Literature review

Before understanding what crypto wallets are and how they work it is vital to understand the underlying technology Blockchain and some of the other key concepts like block mining, addresses and hashing.

2.1. Blockchain

Blockchain is a decentralized and a distributed ledger. It is decentralized because there is no single or centralized authority that looks after the chain. It is called distributed because there are multiple copies of the chain. It can be programmed to not only record financial transactions but everything of value. There can be a Blockchain to record property transactions or any kind of transactions that have economic value. The sole purpose of the technology is to allow fast, secure and transparent peer-to-peer transactions.

A traditional transaction relies on an intermediary for the transaction to take place. For example, if A wants to send money to B, Then A would contact his/her bank to initiate the payment process. In this example bank is the intermediary. It can take a couple of days for the successful transfer of money thus making the whole process time consuming. Whereas, if we adopt the Blockchain technology then it is a matter of minutes as no intermediary is present.

Blockchain can be considered as a digital database that is shared by a large number of computers in the network. These computers are often referred to as nodes. The fact that every node has a copy of the Blockchain provides redundancy which guarantees fault tolerance. In our previous example where A transferred money to B via an intermediary, there is only one authority involved that stores the transaction data or in other words there is just a single source of truth. If the intermediary corrupts the data, then there is no way to check against the integrity of the data. Blockchain solves this problem as every node has a copy of the Blockchain and even if a node tries to corrupt the data then the chain would be deemed unfit as majority of the chains in the network would not match with the corrupted one.

Blockchain is composed of blocks linked together (See Figure 2.1). Blocks 10, 11 and 12 are a part of the Blockchain and are linked. Every block in the chain has a unique ID. Each block has a
field called Prev.Hash (previous hash) which is nothing but the ID of the previous block. Hashing is explained in detail in the cryptographic hashing section. Each block is in turn composed of various transactions. A transaction is nothing but the data associated with a transaction. Every transaction also has a unique ID called as transaction ID to uniquely identify a transaction. Block 11 (See Figure 2.1) has a Tx_root field, which is the root of the tree comprising of the hashes of the transactions. The transactions are hashed and the hashes are in turn hashed until the root node which is Tx_root in the figure. This tree is called Merkle tree of transactions in each block [1]. This is the data structure for storing transactions in each block. It is done to make the integrity check of the transactions simple. Even if a single transaction is altered in the tree then the hash of the root node would change, which would in turn change the hash of the block. If the hash of a block changes then the hash of the following blocks will also change and the whole change would be regarded as a forged one by the nodes in the network.

The first takeaway from the diagram below (Figure 2.1) is that a Blockchain is a linked chain of blocks. The second takeaway feature is that this linking property of the chain provides security. Even a minute change in any of the blocks would corrupt the chain.

![Blockchain diagram](image)

*Figure 2.1: Blockchain consisting of blocks linked to each other*

**Addresses: Cryptocurrency wallets are associated with addresses**
To transact for cryptocurrencies every wallet must have at least one pair of keys. These keys are mathematically derived and are related to each other. The pair comprises of a public key and a private key. The public key is derived from the private key. Private key is used to get access to the funds that a user owns. Getting access to funds is the same as getting the right to spend the funds. Private key can be considered as the password of your credit card. If anyone gets hold of it then they can transfer your funds. This is the reason why it is named private and is therefore meant to be kept safe. On the other hand, the public key is public and is meant to be shared with people. Public key can be considered as the credit card number or the bank account number. The sender must know the recipient’s public key to send funds correctly.

The key point here is that every transaction must be associated with a key. While sending funds the sender sends it to the public key of the recipient. The funds then get locked to the public key and can only be accessed by the corresponding private key.

**Block Mining**

Bitcoin network is a network of nodes. These nodes are nothing but powerful computers that run the bitcoin software. They ensure smooth running of bitcoin by participating in the relay of information. A node is connected to other nodes which are further connected to other nodes and so on. A node will relay information to the nodes it is connected to, who will in turn relay information to the nodes that they are connected to. This is how data propagates across the network.

Out of these nodes some are referred to as mining nodes. These nodes group the transactions that are validated by the network into blocks and add them to the Blockchain. This act of adding blocks to the chain is known as mining [2]. To form a block, miner will have to solve a complicated mathematical puzzle [2]. The first miner to solve the puzzle will announce it to the network. After this all the other miners will stop trying to solve the puzzle for the current block and will start solving the puzzle for the next one.

A cryptocurrency transaction only becomes a part of the Blockchain if is mined by a miner.

**Cryptography: Securing communication**

Cryptography is the practice or study of techniques to secure information in the presence of attackers. It is the math in cryptography that makes Blockchain secure. This section will highlight
some key concepts that will be useful for wallet development. The section will cover public key cryptography used for digital signatures and cryptographic hash functions.

2.2. Public key cryptography

If people want to communicate securely over the internet, they can use asymmetric key cryptography. This technology comprises of two keys mathematically related to each other. The two keys are often referred to as public and private keys. The public key is derived from the private key. The main use of public key cryptography from Blockchain’s perspective is to create a digital reference about the identity of a user. In other words, it provides a way for a user to get identified on the bitcoin Blockchain network. It is like a fingerprint of the user.

2.3. Cryptographic Hashing

Hashing can be regarded as the backbone of the bitcoin Blockchain technology. It prevents data tampering and thus ensures data integrity.

Hashing is the process of converting huge amounts of data into a fixed length of data [3]. Bitcoin network uses SHA-256 (Secure hash algorithm). The 256 refers to a 256-bit output or digest which is generated as a result of passing input data to the function. It is a one-way function; original data cannot be retrieved from the hash.

2.4. How does hashing provide data integrity?

One of the most important property of hashing is that even a tiny change in the input changes the hash completely.

SHA-256 of “I love cryptos” is

“5654AC9DEBAF9A20CF19E74C4A27181E6789D7BE1973419BC0C0FF9CE4175CAF”

What will happen if “I love cryptos” is changed to “We love cryptos”. We can the following hash

“23E405F9E5AE10DB3E1725280D2A90B987C9643FDBA6C8CB75563C59F2BDEB95”, which is a completely different hash.
This property of hash functions provides data integrity and from bitcoins perspective is useful for the following four reasons:

1. Encoding wallet addresses
2. Encoding transactions
3. Validating and verifying account balances
4. Mining
5. Validating the Blockchain

2.5. Digital Signatures

Digital signatures are similar to handwritten signatures and are used to verify the user. Digital signatures are analogous to finger prints in way that once signed the user cannot later claim that it was not done by him/her. Blockchain makes use of this technology to sign transactions.

2.6. Signing and Verification: How digital signatures are verified

1. Sender creates a hash of the transaction.
2. Sender signs the hash using his/her private key hence forming the digital signature.
3. The signed hash is received by the recipient together with the public key of the sender and the transaction data.
4. The recipient takes the data and generates the hash again of the transaction data.
5. The recipient uses the sender’s public key to decrypt the digital signature and retrieve the hash from it.
6. The recipient finally compares the hash generated in step 4 with the one in step 5 and verifies that the sender owns the cryptocurrencies.

Every cryptocurrency transaction is first signed using the senders private key before broadcasting it to the network

2.7. Cryptocurrency: An application of Blockchain

It is a virtual or a digital currency designed to work as a medium of exchange. The first most popular and widely accepted cryptocurrency is Bitcoin and was invented by Satoshi Nakamoto in the late 2008. Satoshi called it a “Peer-to-Peer Electronic Cash System” [5]. It is called a peer-to-peer system because there is no central authority involved. Cryptocurrencies can be directly transferred from the sender to the recipient.
Cryptocurrency is an application of the Blockchain technology as it leverages the concepts of Blockchain. Cryptocurrencies can be used to pay for hotels, flights etc.

The only way that cryptocurrencies are different from physical notes or other currencies is that it is digital. In terms of financial value it is as valuable as a bank note and can be exchanged for it.

**Cryptocurrency wallet: A platform to trade for cryptocurrencies**

It is a digital wallet used to send and receive cryptocurrencies. Unlike real wallets crypto wallets do not actually store the currencies. It just stores the private key (known only to the wallet owner or the wallet service provider) and the public key which has cryptocurrencies associated to it. Apart from acting as an interface for sending and receiving funds it also acts as a personal ledger of transactions.

### 2.8. Wallets in the market

There are a variety of crypto wallets in the market which can be classified into the following five types [6]:

1. Web
2. Mobile
3. Desktop
4. Paper
5. Hardware

Crypto wallets can be either hardware or software wallets. These wallets differ based on how they manage the public and private keys. Some wallet firms give full custody to the owner of the wallet to keep the keys safe whereas some keep the keys and transact on behalf of the owners. Both techniques have its own advantages and disadvantages.

**Web wallet**

Web wallets can be accessed via different browsers such as Firefox, Chrome, IE etc. Web wallets are hot wallets because they are always connected to the internet. Web wallet implementation can vary based on where the keys are stored. Some firms generate the keys for the user and delete them after generating while others save the key pairs in their databases. Web wallets can be hosted or non-hosted. Some of the popular web wallets are BitGo, BTC.com and Coin.Space.
Mobile wallets

Mobile wallets are the most popular wallets in the market because of the convenience they provide to the user. They are hot wallets as they are connected to the internet and therefore safety is a concern. It is easy to find mobile wallets for popular cryptocurrencies that are compatible with both android and iOS platforms. Some of the popular mobile wallets are Mycelium and Coinmi.

Desktop Wallets

Desktop wallets are standalone applications. Such executables can be installed on a machine without having other dependencies. These executables are compatible with majority of the operating systems like Windows, Linux and Mac. As these applications are installed on a machine that is often connected to the internet, one must install antivirus on the machines to prevent theft of keys. Some of the popular desktop wallets are Exodus, Electrum and Bitcoin Core.

Paper Wallets

Paper wallets are extremely safe as compared to other hot wallets as they have the private and public keys printed on them. Paper wallets are secure, but one must be careful about not losing them. The main advantage of using a paper wallet is that it keeps the private keys offline. The only concern is that of the paper quality. The process of generating a paper wallet involves printing the private keys and addresses onto a paper.

Hardware wallets

Hardware wallets are physical devices. These wallets are only built to store the private keys and public addresses. These wallets are also known as “cold storage” wallets as they are not connected to the internet. These wallets are the most secure wallets in the market. When a user wants to broadcast a transaction to the network they momentarily connect the device to just sign the transaction. Some of the popular hardware wallets in the market are Ledger Nano S and Trezor.

After going through the types of wallets that exist in the market we can conclude that hardware wallet is the most as it is not connected to the internet, hence minimizing the chance of being a victim of a cyber-attack. We can also conclude that online wallets like wallet web applications are less secure as they are online wallets operating on the internet (world wide web) and are thus more prone to cyber-attacks. Despite of the less security that web wallets offer they are still more popular
as they are easily accessible over the web. Anyone having an internet connection can sign up for a web wallet.

This project will try to integrate sophisticated security features like two-factor authentication, generation of hierarchical deterministic seeds for making the key management less error prone and multisignature feature to the online web wallets. Adding these secure features to the web wallets will add an extra layer of security and thus will be a step towards making them widely acceptable amongst the investor community.

In the next section, the report will discuss the development methodology of the web wallet service. It will explain the architecture of the web application. It will then provide a component wise breakdown of the web application along with justifications.
3. Methodology

This section will explain the architecture of the application followed by a component wise description of the architecture.

3.1. Architecture

![System Architecture](image)

*Figure 3.1: System Architecture comprising of client, server, database and 3rd. party API*

The application architecture comprises of a client/front-end, a server API, a database and a 3rd. part API that the server API will communicate with to connect to the Blockchain network.

3.2. Components

The system architecture follows basic client-server interaction model. The client will interact with the server and the server in turn will interact with other 3rd. party application programming interfaces to connect to the testing Blockchain. The third party API used by the application will be BLOCKCYPHER’S developer’s API. The server will also interact with the database to store/retrieve user information (See Figure 3.1).
3.2.1. Front-end

The front-end or the client side of the application will be rendered by the backend server implemented using Node.js and Express framework. The rationale behind selecting Node.js and Express is that it has a lot of support modules for enabling features like two-factor authentication and seed generation for hierarchical deterministic wallets.

3.2.2. Server-side

The server side of the client application will use Express.js framework. The application will be implemented using Node.js which allows JavaScript code to run outside the browser. There are many Node.js frameworks in the market but this app will be using Express.js as it provides greater flexibility. Express integrates easily with other modules to get the desired results.

3.2.3. Database-side

The database side of the application will be implemented using a NoSQL database. The reason behind choosing a NoSQL database is that it is less rigid as compared to other relational databases in terms of data storage techniques and thus will provide greater flexibility. The database will be hosted on cloud. The application will mLab which is a cloud database service that hosts MongoDB databases.

JavaScript is chosen as the language for the server side code as there is a plethora of bitcoin libraries available to implement features like two-factor authentication and also Blockchain related functionalities.
4. Features

This section of the report will go through all the security features that our wallet will possess. For every feature the report will explain how it will be integrated with the wallet application and how it will enhance the security of the wallet. This section will emphasize more on the technicalities of two-factor authentication, Multisignature wallets and Hierarchical deterministic wallets.

4.1. Two-Factor Authentication

Two-factor authentication in conjunction with the traditional authentication measures like providing the username and password will be used to grant access to the wallet account. This will ensure a higher level of security as during login process user will be asked to input a one-time password which will change in every 60 seconds. This OTP will be generated by Google authenticator.
How it works?

1. Open the mobile app store or play store and install “Google Authenticator” (See Figure 4.1) and install it.

2. After registering for the wallet account on the web application the user will be prompted to scan the displayed QR code. To scan the QR code open the Google Authenticator app and click on “scan a barcode” (See Figure 4.2) [9].

3. After completing the scanning part, the authenticator app will generate a pin that will expire and regenerate in every 60 seconds. The user will be prompted to enter this pin during the login process (See Figure 4.3) [9].
Passwords can be hacked and OTP’s (One-time passwords) generated to reset the passwords can also be sniffed but two-factor authentication requires a person to be in possession of a physical device. This feature makes it impossible for a hacker to get access to the account without possession of the physical device. This will make the wallet secure by adding an extra layer of security to the login step.

Two-factor-authentication will be implemented using a 3rd party JavaScript package called Speakeasy. It will be installed using npm (Node package manager). It is a onetime passcode generator that is ideal for two-factor authentication and supports two-factor devices like Google authenticator.
Application UI for two-factor authentication

Step 1: Enter account credentials

The above UI is for accepting the username and password. These credentials are what the user knows.
**Step 2: Enter authentication code generated by Google Authenticator**

After the user successfully authenticates by providing the username and password. The next step requires the user to open the Google authenticator app installed on the device which generates an OTP or code. This OTP is what the user physically has.

This cannot be hacked as for hacking the hacker must be in possession of the physical device which is the phone in this case.

Thus we have been successful in providing a two level authentication to our users.
4.2. Multisignature Wallets

This section explains how cryptocurrency transactions work and then discusses how multisignature feature can make the transactions more secure.

How the application uses this concept?

Address Generation

A multisignature transaction is signed by multiple private keys. Our application requires the user to first create a multisignature address by selection and two addresses from the list of addresses ever used by the user.

![Select two addresses to generate MultiSig Address](image)

We can see from the above UI screenshot that the application requires the user to select two addresses because the private key of these two address will be used to sign the transactions created from the multisignature address.

After the user clicks the “Generate Address” button these two address are sent to the back-end and from there are sent to the BLOCKCYPHER’S API for address generation.
Spending from a multisignature address

In order to use multisignature addresses we first need to fund them. These addresses can be funded like any other address. The user can transfer funds from a non-multisignature address to a multisignature address and from a multisignature address to another. After the multisignature address is funded we can now spend. The key for the addresses that we use during the multisignature address generation are stored in the database. Now we have to send the public keys user to generate the address, script type, destination address and the value to the blockcypher’s API. The API returns an array of data to be signed by the private keys. If the script type, we use is multisig-n-of-m then we will require n signatures for every element in the array which has to be signed and returned. After completing the signing process we return the signed data back to the API for approval and broadcasting to the network.
4.2.1. Transactions

Cryptocurrency transactions work like any other digital currency transactions. For a user A to send money to any other user, the transaction must be signed. Every user has a public key and a private key. Public key as the name suggests is public and can be broadcasted to the network. User A to receive cryptocurrencies from any other user must provide that user its public key whereas the private key is supposed to be kept safe and not to be shared with anyone. To broadcast a transaction on the network one must digitally sign the transaction. This signature is made using the private key of the sender. Once signed, the sender later cannot claim that the transaction was not signed by him/her. Therefore, it is the responsibility of the sender to keep the private key safe because once a transaction is made it cannot be reversed.

Now let us discuss some possible problems with this transaction model

1. What if the user by mistake makes a transaction?
2. What if someone else gets the private key and makes the transaction?

To mitigate the risks involved with simple transactions we introduce the idea of Multisignature wallets. Multisignature is an idea used to add security to the process of how transactions are carried out.

Let us refer to figure 4.4 to understand how Multisignature technology works. We can notice in the figure that there is a locker and two key insertion points on it. Let us assume that user A has some money in the locker and wants to take out some of it to give it to user B. User A currently has just one key for the locker and the other one is kept safe somewhere else. To open the locker user A needs two keys. This is the point where the whole idea of using multiple keys makes the system secure. If an attacker gets one of the keys, the other one will also be needed to open the
locker. Now one can ask what if the attacker gets the other key also. In such a scenario the attacker would be able to steal all the funds. This technology does not completely rule out the possibility of a theft, but it makes it harder for the attacker to retrieve multiple keys kept at different locations. This in turn makes it difficult to transfer funds from one account to another.

This is how a Multisignature transactions work. It requires M-of-N signatures to broadcast a transaction [8]. If M =5 and N=3 then at least 3 signatures are required to broadcast the transaction to the network. It is advised to have a higher value of N as it makes it difficult for the attacker to steal all the keys [8]. Our crypto wallet will implement Multisignature feature to enhance the security of the wallet.

4.3. Hierarchical deterministic wallets

One of the main concerns of wallet users is managing private and public key backups. Hierarchical deterministic wallets allow users to generate all the private and public keys from a single point. This single point is known as a seed. Without HD wallets users must keep the records of all the key pairs separately. Managing multiple backups is error prone and not safe at all. HD wallet allows the user to regenerate all the key pairs just from a single backup that serves as a continuous and a seamless backup. Thus, HD wallets are a step towards enhancing the security of the wallets. The wallets are said to be deterministic because the private key generation method is deterministic. In normal crypto wallets the key generation method is random. Because of the randomness in the key generation process there is no link between the generated keys. This has a major disadvantage when compared to the deterministic style of generating keys and that is the user will never be able to re-generate the keys in case of theft or loss. If the user cannot recover the keys, then it is impossible to regain control over his/her funds.

This section of the report will explain how HD wallets work in more detail by providing a stepby step illustration of how infinite number of related keys can be generated from a single seed.
Steps to generate deterministic keys [7]

1. Create a random master seed (See Figure 4.5). This seed is a phrase which has words taken randomly from a dictionary. It is usually a twelve-word phrase [7].

2. From the seed the system generates the root public key (See Figure 4.5 Depth=0). The root public key generated in step 2 is the root of our key tree [7].

3. From the root we generate our first address node by selecting the right wallet account and wallet chain (See Figure 4.5 Depth=3.5). The address m/0/0/0 is the first address and m/0/0/1 is the second and it goes on till m/0/0/k [7].

4. The generated addresses are converted to bitcoin address format.

An infinite number of addresses can be generated using this technique and the key characteristic of this technique is that we can generate the same set of keys if we start with the same seed hence making this key generation technique deterministic. This feature will the make the wallet more secure as the user would not worry about managing large sets of keys and would instead just safely store the seed.

There are existing packages that will be used to generate related addresses from a mnemonic (12-word phrase). These packages will be installed used the node package manager (npm).
5. Application functionalities

1. **Send Satoshis**- This feature allows the user to send satoshis from an input address A to an output address B.

   From: Lets the user select the address from the list of addresses ever generated by the user.

   To: Input the recipient’s wallet address.
2. **Generate a wallet address** - This feature allows the user to generate a new address.

This address is under the hood generated using the mnemonic and thus is related to all the previously generated addresses.

After clicking on “Generate Address” button the UI shows the generated address. The newly generated address is funded with 1000000 Satoshis.
3. **Check balance of a wallet address** - This feature allows the user to generate a new address.
4. **Request Satoshis**- This feature allows the user to copy the address to the clipboard. This feature allows the user to correctly copy the address with any errors.
5. **Generate MultiSigAddress** - This feature allows the user to generate a new multisignature address. The user will have to choose two addresses whose private keys will be used to sign the transactions from the generated multisignature address.

6. **Check multisig address balance** - This feature allows the user check balance of a multisig address.
7. **Send from multisig address** - This feature allows the user to send satoshis from a multisig address.
8. **Retrieve address used using the mnemonic** - This feature allows the user to generate a list of addresses ever used by the user using the mnemonic given to the user at the time of account creation.
6. Difficulties Encountered

This section discusses the difficulties encountered in the development process.

The application will be integrated with a testing chain and not with the main Blockchain. The major benefit of using a test chain is that real money will not be required to carry out the transactions. There are a lot of protocols compatible with the main chain and not with the testing chain. These protocols are rules that I will have to adhere to make my wallet compatible to work with bitcoins. There are some challenges with respect to making the protocols compatible with the testing chain. The keys (public and private keys) used for main Blockchain and testing chain are generated using a different set of algorithms. There are currently no API’s in the market for generating wallet addresses for the testing chain of the BLOCKCYPHER’s API. Generating wallet addresses from scratch involves in depth understanding of hashing algorithms and encoding methods which is a challenge as I am not familiar with any of the hashing algorithms. There are some 3rd party libraries that I will be using to implement hierarchical deterministic wallet and multisignature technology. Understanding how to use these 3rd party libraries will be a challenge as I will have to understand how exactly the library works. There are some challenges with respect to making the protocols compatible with the testing chain. Generating wallet addresses from scratch involves in depth understanding of hashing algorithms and encoding methods which is a huge challenge as I am not familiar with any of the hashing algorithms implementation libraries.

There was some difficulty in understanding how related addresses will be generated for the hierarchical deterministic wallet. As infinite addresses can be generated by changing the derivation path. We needed to come up with a way to have a link between every generated address so that we could generate all the used addresses again.

Some of the other challenges that I am currently facing are that I have no prior experience with making an Express app and deploying it on cloud or hosting service providers. This application will have a micro services architecture and I am not familiar with the designing aspect of such an architecture.
7. Project progress

The development, deployment and testing of the application has been completed and the application is live on https://safe-ravine-61737.herokuapp.com/.

What has been implemented so far?

A login system for the web wallet app with two-factor authentication has been implemented. Every time a new user registers for a wallet account he/she is asked to scan the QR code into the Google Authenticator app which will generate a one-time token that will be needed during the login step apart from the username and the password. After a user registers for an account a secret is generated and is saved next to the user data in the database. This secret is convert into a QR code which the user is prompted to scan. After successful login the user lands on the dashboard page.

I have also implemented the hierarchical deterministic wallet. Upon successful registration and login, the user will be asked to save or write the mnemonic phrase somewhere and store it in a safe place. This phrase will be used to generate all the linked key pairs that will be used for all the future transactions. There will be no need to save the key pairs and also create backups for every key pair. Unlike nonhierarchical deterministic wallets our wallet will just make use of a 12-word mnemonic phrase to generate the kay pairs that will ever be used by the user.

I have also completed the multisignature address feature. The application requires the user to select two addresses for generating the multisignature address. In other words, it is a script type of 2 out of 2 which means that the system will require two private keys to sign the transactions. The two addresses that are used to generate the multisignature address will have two corresponding private keys which will be used to sign the transactions made by a multisig address.
<table>
<thead>
<tr>
<th>Tasks</th>
<th>Status</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial research</td>
<td>Completed</td>
<td>September-October</td>
</tr>
<tr>
<td>Architecture design</td>
<td>Completed</td>
<td>October-November</td>
</tr>
<tr>
<td>Development</td>
<td>Completed</td>
<td>November-March</td>
</tr>
<tr>
<td>Testing</td>
<td>Completed</td>
<td>Mid-march</td>
</tr>
<tr>
<td>Deployment</td>
<td>Completed</td>
<td>April</td>
</tr>
</tbody>
</table>
8. Future Steps

1. Add other cryptocurrencies to the wallet.
2. Add a messaging system so that we can request for transaction signing from other peers.
3. Adding customer support feature.
4. Having a mobile app for both iOS/Android
5. Accounting feature for the past transactions.
6. Adding a feature for dynamic fee suggestions.
7. Adding tor for additional privacy.
9. Conclusion

This report has discussed how crypto wallets came into being. It has also emphasized on the need of making the online wallets secure as they are very popular amongst investors. This report explores three features that can be implemented to make the online wallets more secure. The first feature is two-factor authentication. It is shown how requiring a physical device to generate a temporary pin in addition to what the user already knows which is the username and the password makes the wallet secure. It then explores the second security feature; hierarchical deterministic wallet and shows how management of a single seed makes the whole process of key management less error prone and thus more secure. Finally, the report also discusses about how the multisignature technology requires multiple keys to sign a transaction. The requirement of multiple keys makes it tough for an attacker to get access to all the keys, thus making the wallet secure.
10. References


