EXAMENSARBETE INOM TEKNIK,
GRUNDNIVÅ, 15 HP
STOCKHOLM, SVERIGE 2018

Comparison of blockchain e-wallet implementations

Behnam Eliasi

Arian Javdan

KTH ROYAL INSTITUTE OF TECHNOLOGY
ELEKTROTEKNIK OCH DATAVETENSKAP

Abstract

With the rise of blockchain technology and cryptocurrency, secure e-wallets also become more important. But what makes an e-wallet secure? In this report, we compare different aspects of e-wallets to see which alternatives are secure and convenient enough to be used.

This report contains comparative analyses of different implementation for e-wallets. The problem area is divided into three smaller areas: Key storage, authentication, and recovery. These problem areas have defined criteria for what is considered good qualities in each respective area.

The results show that for key storage, the best options are, Android's keystore/IOS' secure enclave, offline storage or a hybrid hot/cold storage. For authentication, the best alternatives proved to be BankID and local authentication through the phone's OS. Good Recovery alternatives include recovery seeds that recover the whole e-wallet or using multiple keys for both signing and recovery.

The proof of concept made for this project uses three different storage methods with the authentication methods for each one and with the possibility of recovery in case a key should be lost. The storage methods used are offline storage thought QR-codes, online storage with firebase and local storage with Android keystore or Secure enclave. Authentication is done with Facebook/Google sign in or local authentication.

Keywords: Blockchain, key storage, mobile payment, e-wallet, authentication

Sammanfattning

Med blockkedja och kryptovalutornas ökande popularitet blir säkra e-plånböcker allt mer viktiga. Men vad gör en e-plånbok säker? I detta arbete ska olika implementationer för e-plånböcker undersökas för att se vilka alternativ som är tillräckligt säkra samt användarvänliga.

Problemområdena delas upp i följande delar: nyckellagring, autentisering och återhämtning av stulen/förlorade nycklar. Arbetet innefattar jämförelser mellan olika lösningar till dessa områden med definierade jämförelsekriterier.

Resultatet visar att för nyckellagring är de bästa alternativen Androids keystore system/IOS secure enclave som båda är en form av säker lagringsplats på telefonen, offline lagring och hybridlagring som enkelt förklarat är en tjänst som bevarar data offline och gör den online när användaren väl vill ha tillgång till datan. För autentisering är de bästa alternativen BankID och lokal autentisering genom telefonens operativsystem. För återhämtning av nycklar är de bästa alternativen recovery seed eller att använda multipla nycklar för både signering och återhämtning.

En proof of concept gjordes där lagringsmetoderna papper (exempelvis QR-kod), online-lagring med Firebase och lokal lagring med Android keystore eller Secure enclave implementerats. Autentiseringen sker med hjälp av Facebook/Google login och lokal autentisering. Återhämtning görs med två utav tre nycklarna som används för både signering och återhämtning.

Nyckelord: Blockkedja, nyckellagring, mobilbetalningar, e-plånbok, autentisering

Table of contents

1 Introduction

This project was done in cooperation with the Swedish Blockchain Company, Centiglobe with the goal of finding suitable ways to make a blockchain e-wallet [1]. This section will introduce the problem, the goal and the limitations that were set for this project.

1.1 Background

The process of making financial transactions includes many banks which can take several working days. The Swedish Blockchain company, Centiglobe, has a goal to make bank transactions faster, more secure and open for others to implement. This will make it possible to work with a decentralized blockchain instead of using the bank systems which is being used today.
Centiglobe is currently working with a Blockchain system that uses three keys. Two of these keys are used for authenticating the user to the blockchain when signing a transaction, and the third key can be used as a back-up key in case one of the other two should be lost. The issue here, and with Blockchain in general, is how to store the keys since if they are handled incorrectly the funds connected to those keys are at risk.

1.2 Problem

The problem with handling blockchain keys is the risk involved. If the keys are stolen or observed, then the user can lose access to all their funds; for this reason, e-wallets are used to make this process easier for the user. Since the use of blockchain technology is on the rise, it can be relevant to look at the different methods available for making such an application. But what makes a good e-wallet? This question can be divided into three sub-questions:

1. How do we suitably store blockchain keys?
2. How do we authenticate ourselves towards the storage method in a suitable way?
3. How do we recover part of or a whole e-wallet?

1.3 Purpose

The purpose of the project is to make an application for mobile payments via blockchain in a secure fashion that is also convenient for the user. The project includes the process of making the service and the decisions made along the way. The idea is to make an application that can be used easily by users who are not tech-savvy or do not have experience with blockchain.

1.4 Goals

The work in this project is divided into three categories: Storage, Authentication and Recovery. For each of the listed areas, a comparison will be conducted based on some evaluation criteria to see which of the methods make for good alternatives and could be used in an e-wallet of this type.

Storage has to do with the storage of the blockchain keys if there are multiple keys different storage methods should be chosen for the keys. The storage methods are evaluated based on security criteria listed in section 3.2.

Authentication has to do with how the user should authenticate their identity to gain access to their blockchain key. The evaluation criteria for authentication are listed in section 3.3.

Recovery concerns the scenario when the user has lost a key or if the key has been stolen in some way. The evaluation criteria for this are more about looking at what recovery possibilities exist at loss/theft.

Something also must be said about the convenience of the application. Purely from a security perspective, it might be a good idea to store the blockchain key on an offline device that can sign a transaction and then move the transaction from the offline device to an online device that can then send it to the blockchain, but this is likely too cumbersome a process for the average user. Hence, the convenience of the application is a factor that needs to be included, which is something along the lines of the number of steps required to access the key.

As for the application, the goal is to make a proof of concept where a user can choose from different storage methods for their keys where each of the storage methods has at least one authentication method mapped to it. Once the user has chosen their preferred storage methods and stored their keys there, it should be possible to sign transactions and recover lost/stolen keys through the app. To sign a transaction, the user needs to access their keys, meaning they will need to provide authentication for each of the keys to be accessed. To recover a key, the user will need to provide the necessary credentials (more on this in the section about Recovery). Once these credentials have been provided, the user should choose one of the available storage methods, and a key will be created there.

1.5 Research Methodology

The bulk of the project is going to be a series of comparative analyses for finding appropriate methods for an ideal multi-key e-wallet; this will then also be adapted into a proof of concept application.

The programming language that is going to be used for writing the application is react-native. React-native is suitable because it allows for the development of an app which can run on both Android and IOS devices [2]. React-native also gives the opportunity to test-run the application directly on the mobile phone (Both IOS/Android) without any difficulty.

1.6 Delimitations

This report will only concern itself with the goals mentioned in section 1.4 and by extension, the factors mentioned in 3.2, 3.3 and 3.4. This report will not concern itself with the graphical aspect of an e-wallet application but rather the functional aspect, i.e., the implementation.

1.7 Ethics and Sustainable Development

Ethical aspects in the blockchain are important where it is important that the users' information is anonymous and can’t be seen by others. The application is going to be used on mobile phones, and therefore it's very important that user information does not leak. It should not be possible for a user to sign in with the same login session. This is for security purposes where the user can lose their device, and the thief should not be able to access keys from an old login-session.

Social aspects that are going to be affected are that people are going to be able to transfer money easily without the need for a central authority. The blockchain system makes it possible for people that have no access to banking services like debit cards, e-wallets or ATMs to make transactions all over the world. This makes it possible for anyone wherever in the world to make transactions to anyone without exceptions. It makes it possible for people who don’t trust their banking services in their country to make transactions without the untrusted central authority.

Environmental aspects from the blockchain are especially energy consumed for being able to use the service. When a transaction is done all in the blockchain has to confirm the transaction, which means that there is a need for energy for holding the whole blockchain online all the time. Imagine a lot of transactions being done in the future where thousands of users at the same time need to have their blockchains signed where this results in a lot of electricity being consumed.

Economic aspects of blockchain are that the users of blockchain will decrease their fees during transactions. Most of the bank services have a different amount of fees when doing transactions, and this will be avoided when using blockchain.

2 Background

This chapter will shortly explain the history of blockchain and its use. The chapter will also introduce existing technologies that will be used in the comparisons that are to be made in chapter 4.

2.1 What Is Blockchain?

Blockchain is a technology from 1991 which was described by Stuart Haber and W. Scott Stornetta. Blockchain was built to create a secure chain where each block is a series of data, and each of the data is connected to the block before like a chain [3].

In 2008 the unknown Satoshi Nakamoto used blockchain for securing a history of data exchanges by using peer to peer networking each block got a timestamp from each exchange and make it possible to verify each of the transactions. All this could be made without any central authority like banks, which was the start of the Bitcoin which use blockchain technology [3]. Instead of using central authorities, blockchain uses decentralised ledgers for trust [4].

By using blockchain without a central authority like a bank, the user will avoid different kinds of chargers from a bank. Most banks take some charges for transactions per year where different type of services cost at different prices.

Blockchain will decrease the delays that occur because of conflicts/confusions in financial services, duplicated information and banks confirming transactions [5]. This kind of delays often occurs during international transactions that can take days until they are completed. If we instead use blockchain, the transactions can be done in a matter of seconds compared to the traditional financial services.

Blockchain includes the ability to trace all the transactions since all blocks are in chronological order. Each block is connected to its neighbouring blocks, which are cryptographically hashed. These abilities make it easy to track and examine the block information [4].

2.2 Authentication

This section will introduce two different forms of authentication methods, offline authentication and online authentication. An online authentication is a form of authentication that is done through an internet service. Common examples of this are Google and Facebook login that will be explored alongside BankID, a Swedish e-authentication system. Offline authentication, unlike online authentication, is restricted to some form of local authentication of which there are different technologies that can be used, the ones covered in this report are Face recognition, fingerprint recognition and PIN/passcode.

2.2.1 Local Authentication

Local authentication is an authentication method that is done locally on the devices' operating system. Examples of this are the pin code/password, fingerprint or face recognition system that is being used

for accessing a mobile phone. Local authentication can be used for more than just letting the user access their mobile phone. It can be used for many different applications like bank applications, social media and more. Therefore, it could be an alternative to using local authentication for accessing keys. We will evaluate the authentication method by assessing face recognition, fingerprint, PIN and password authentication separately.

2.2.1.1 Fingerprint Recognition

Techniques used for fingerprint recognition on all mobile phones are similar where the sensor creates a small picture of the finger. When the user each time want to authenticate themselves their finger picture gets compared to the saved fingerprint in the mobile phone. The saved fingerprint is the fingerprint the user used when they configured their phone.

According to mobile companies, their fingerprint is very safe to the point where they are being used for bank transactions; for example, Apple Pay / Samsung Pay. According to Apple, somebody else's fingerprint will be accepted in a system one time in 50.000 [6]. However, there are more significant problems with fingerprint authentication. According to new research done in New York University & Michigan State University, the fingerprint may be less safe than the companies claim [7]. Fingerprint sensors in mobile-phones have prints from a small part of the fingers, due to this, the researches could easily fool the sensors with the help of fake fingerprints digitally composed of common features found in human fingerprints [7]. With the fake fingerprints, they could get a match 65% of the time. Although the test wasn't done on mobile phones, the researchers claim the test still means the number provided by Apple (1 to 50.0000) should be taken with a grain of salt. The problem is also that people tend to have several fingers on their phone and that the mobile phone takes several pictures of their fingers. This also makes it easier for a person with a fake fingerprint where they only have to match with one of those pictures of fingerprints [7].

2.2.1.2 Face Recognition

There are three different techniques available today for face recognition, which is 2D face recognition, 2D-3D face recognition and 3D face recognition [8].

2D face recognition technology starts by trying to detect if there is a face in an image or not. Where the recognition system usually can determine if there is a face on an image or not. When a face has been detected, the system starts with taking out different features which are in the face [8]. This is the part of the system which finds the uniqueness in the faces and creates a signature from the faces. When the uniqueness is found, the system goes to the authentication parts where it compares to faces which are stored. The only technical device used is the camera for creating an image that the faces gets compared with. The disadvantages with 2D face-recognition are that it's not reliable enough for being used for security [9]. There have been mobile-phones which have used 2D Face recognition where people easily could trick the system with photography [10].

3D face recognition technology offers a more precise recognition of the face, which creates a 3D image of the face [8]. With the help of using more 2D cameras, a 3D image can be created. Technologies like infrared or laser sensors have also been used in some devices for making the 3D face recognition more precise [11]. This is an expensive technology which is mostly used in places where high security is needed although there are some implementations in mobile phones where cheaper techniques of 3D face recognition are used for authentication. The phones usually use infrared combined with a 2D camera with a projection which creates a 3D image of the face [11]. The implementations are safer than 2D face recognition but still have their security flaws. Apple, who uses a 3D face recognition device, claims that only 1 of 10000000 can unlock another device [12].

2D and 3D face recognition technology together are when some parts of the 2D technology and some parts of the 3D technology are combined [8]. This implementation is used in mobile phones; for example, iPhones FaceID where a combination of both technologies shows better security against the 2D-technology only. In the 2D technology, users could quite easy trick the system with a picture, but that's not the case when both the 3D & 2D is used. There is although still some security risks with this implementation too.

2.2.1.3 Pincode/Password

The pin code & password authentication integrated into the operating system on mobile phones can be used for more things than just locking up the phone. It can be used for authentication on different applications where the pin code/password is device-specific.

2.2.2 Online Authentication

Online authentication is when the authentication part is taking care of third-party service. It can, for example, be Facebook or Google that takes care of authenticating the user to an application where the idea is that users can use their existing Facebook/Google accounts for authenticating themselves. The online authentication method has different alternatives depending on the service provided, but usually, the users have the choice to have device-specific authentication or not.

Another online authentication tool is BankID, which is a Swedish e-authentication service. It was released on 14 April 2010, and it’s being used by many users daily. It can be used for authenticating users to many different services like Banks, authorities, shopping websites and more. The service is device-specific, which means that the user only can authenticate themselves with the device they have configured BankID with. There are no possibilities for a user to authenticate themselves with another device other than the one which has been configured.

The BankID implementation is convenient since 7,5 million of the Swedish population have it on their mobile phones [13]. Most of them have used the service where BankID provides different ways to sign which is by pin code, TouchID/fingerprint and FaceID/face recognition [14]. The user itself chooses which way they want to verify their authentication the only thing that makes BankID less convenient is that the user has to switch to BankID app and back each time they have to authenticate themselves.

2.3 Storage

There are many types of storage methods used in e-wallets, and they can be divided into different categories based on how and where they store the keys. The ones evaluated in this report are hot hosted wallets, cold hosted wallets, a hybrid of hot and cold wallets, local storage and offline storage.

2.3.1 Offline Storage

Offline storage includes methods are that cannot be connected to the internet such as paper wallets (writing the key on a piece of paper), a USB or an external hard drive [15].

2.3.2 Local Storage Device

Local storage is when the key is stored directly on a device such as a mobile phone or a computer. Local storage differs from offline storage since the device can be connected to the internet.
When it comes to local storage on phones, more alternatives are available, namely, Keystore storage. The Keystore storage is an alternative to the regular local storage that has its own storage space and processor to make sure it is separated from the primary storage of the phone while also providing encryption [16]. This is done slightly differently on Android and IOS, but the general idea is to provide a form of secure storage that encrypts data and only decrypts it when proper authentication has been presented [16], [17]. The different encryption techniques to store keys is either asymmetric encryption with RSA, ECC and symmetric encryption with AES, 3DES. The various implementations that exist are Android Keystore system, TrustZone and Bouncy Castle [18].

The Android Keystore system is a system which helps with storing cryptographic keys in a container, which makes it more difficult for other applications to extract the data from it. It protects data from data-sensitive applications from other applications that try to access it [19]. The only application that could access the keys will be the app which is specified with the Android Keystore system other applications will be declined and not see anything of the data which is protected [19].

When an application chooses to use Android Keystore system, the key itself will never enter the application process. The cryptographic process in the applications where plaintext and ciphertext gets verified all happens behind the scenes. Otherwise, if it does not happen in the background, there is a risk that other applications inside or outside the device to get a glimpse of the keys [19].

Depending on the mobile phone, some of them has TrustZone technology implemented in their hardware(ARM processor) [20]. The TrustZone technology implements Trusted Execution Environment(TEE) which bounds the keys to secure hardware and isolate it from the Android OS(Only if a device has TrustZone/ TEE)[20]. It’s protected against loopholes where other systems may have exposed internal storage in the device. Even with access to the internal storage, the person who has found the loophole can’t extract keys from the device [20].

Secure Enclave is a hardware-based key storing system which isolated keys from the main processor for creating extra security. The Secure Enclave does all the actions to keep the key secure and not visible to malware [21]. Secure Enclave is like the Android Keystore System and works like the Android Keystore system. The difference is that the Secure Enclave is the implementation, which is for Apple devices. The hardware implementations are on Apple Devices with an A7 processor(iPhone 5s, iPad Air, iPad Mini 2, iPad Mini 3) and later [21].

2.3.3 Hosted Storage

Hosted wallets are a service that stores the wallet for the user; if the wallet is online, then it is hot; otherwise, it is cold [15]. There are hybrids of hot and cold hosted wallets as well that enables moving funds from the hot wallet to the cold then and vice versa. Hybrid wallets can be useful since it allows for a more significant amount of funds to be kept on the cold storage while smaller amounts can be transferred to the hot wallet to make a transaction [3].

2.3.4 Security Token

A security token is a device which is similar to the devices used for authenticating to different banking services. The difference with Security token and the devices from the bank is that the Security token is a device which apart from authentication also saves keys inside of them.

2.4 Recovery

When a user has lost their keys or has been subjected to theft, the user needs a way to secure their e-wallet, this process will be referred to as Recovery in this project. Recovery essentially means that the user replaces one or more blockchain keys. This has to be done in a way that a perpetrator cannot recover keys he does not own; therefore, the user should provide some form of credentials to reduce the possibility of abuse. In this report, the recovery possibilities explored are recovery seed and multikey recovery.

2.4.1 Recovery Seed

A recovery seed is a list of different words in a specific order where all the words store enough information for restoring an e-wallet [22]. An e-wallet is an electronic wallet where all the users' keys are kept in, so this means that a recovery seed can restore any number of private keys. The process where the recovery seed gets created is when the keys are getting created.

2.4.2 Multikey Recovery

Multikey recovery is possible due to the support of the blockchain that Centiglobe is working with. This blockchain allows for a key to be replaced given some credential, which can be one or more other keys. The blockchain allows for dedicated recovery keys, meaning that you would have one set of keys used for signing transactions and a different set of keys used for recovery but these can also be the same set of keys.

2.4.3 Biometric Encryption

The meaning with this is to encrypt Biometric properties and using them for encrypting keys. Instead of encrypting keys with a password we could do that with biometric encryption. This is not the same as authentication with password or fingerprint; this technique is used for encrypting the keys themselves.

2.4.4 Third Party

An alternative for recovery is to let a third party keep the key, which could, for example, be a bank. It could also be other secure third-party companies which store recovery keys for users. The idea is that the third party keeps the recovery key secure until the user needs it, and therefore, the user does not have direct access to it. This principle applies to both digital and paper storage since it is possible to keep the paper in, for example, safety deposit box at a bank.

3 Method

In this chapter we will define the evaluation criteria for both key storage and authentication while pinpointing the interesting aspects of recovery that needs to be researched. These will later be used as a framework to compare the different technologies in chapter 4.

3.1 Research Process

In this report, three comparative analyses will be done for the areas of Key storage, Authentication and Recovery. For each part, previous work will be examined, such as scientific reports and existing implementations while also taking into consideration how these would work in multi-key solutions. The evaluation criteria for each of the parts are listed in the sections below.

The project also includes a proof of concept that implements some of the technologies discussed in the report to prove that they can be used for blockchain e-wallets and to present one way in which this type of application could be put together. In chapter 5, the proof of concept will be described, and the design choices motivated along with an evaluation of the resulting application.

3.2 Key Storage

When it comes to key storage, there is a study by Eskandari and his colleagues at Carleton University, which defines a framework for analysing key storage methods specifically for bitcoin, which is still relevant to this project. This framework introduces several criteria used to evaluate the storage methods, which are:

- Malware Resistant - A wallet that is not stored on a device with internet access or on a device capable of performing computations is malware resistant.
- Key(s) Kept Offline - Keys not available through the internet are considered to be in offline storage.
- No Trusted Third Party - Concerns whether or not the used tool should be trusted or not.
- Resistant to Physical Theft - If the keys are stored in a way in which they cannot be stolen, they are considered to the resistant to physical theft.
- Resistant to Physical Observation - Physical observation could be for example if the keys were printed on paper they could then be observed if this cannot happen then the storage method is considered to be resistant to physical observation.
- Resilient to Password Loss - If the service uses passwords, can the key be recovered, or can the password be reset if it is lost somehow.
- Immediate Access to Funds – the user has instant access to the keys when using the application and does not have to fetch anything else.
- No New User Software - Does the user have to download any additional software or can the wallet work from a browser?
- Cross-device portability - A key storage method is considered cross-device portable if it can easily share the address of the funds.

3.3 Authentication

The European Central Bank has come up with a set of recommendations for improving security in mobile payments. The recommendations were introduced in the European forum in 2011 with the aim that the European countries should have shared knowledge about the security aspects concerning mobile payments. Therefore, the recommendation sets up requirements that need to be fulfilled for the transactions to be considered safe. These recommendations are in line with the level of security offered by card payments [23].

For authentication of mobile transactions, the bank recommends that two out of the three following credentials be included:

(i) something only the user knows (e.g. a static password, code or personal identification number)

(ii) something only the user possesses (e.g. a token, smart card or mobile device)

(iii) Something the user is (e.g. a biometric characteristic, such as a fingerprint). Also, the elements selected must be mutually independent, i.e. the breach of one does not compromise the other(s). [23]

The bank also notes that the companies or organisation who implement these mobile payment solutions have to be careful in their design to make sure that the authentication is protected. For example, when a user is using their password in service, their password should not be able to be observed by unauthorised people, there should be confidentiality in the authentication. The bank explains further that the data which transferred in the transaction should also be protected such sensitive data like personal information, transaction data, and so on should never come to unauthorised hands [23].

The authentication methods that will be examined in this report are: local authentication such as TouchID and Android equivalents, Google and Facebook authentication and authentication through the Swedish e-identification app BankID. With these authentication methods, we also have to consider other security aspects, such as bypassing of the authentication, brute-forcing the authentication or data leaks.

3.4 Recovery

For evaluating different recovery methods, we look at the possibilities that exist when losing one or more keys. It is also interesting to look at the different outcomes when a key is stolen compared to when it is lost as a key that is lost but not observed might not be as problematic as one who is observed.

4 Analysis

In this chapter, all the different implementations of authentication, storage and recovery mentioned in chapter 2 will be evaluated based on the criteria introduced in the previous chapter.

4.1 Authentication Evaluation

This section will evaluate local authentication and online authentication with respect to the recommendations from section 3.3.

4.1.1 Local Authentication

As mentioned in chapter 2 is an authentication method explicitly tied to the device's operating system. In this section, we will evaluate the authentication method as a whole by assessing face recognition, fingerprint, PIN and password authentication separately.

4.1.1.1 Fingerprint Recognition

First, we look at fingerprint authentication specifically. When it comes to the central bank of Europe's criteria, fingerprint authentication does provide the (iii) rule of being a biometric [23]. Something that does depend on the implementation is the second criteria of including something the user owns. If the application requires the user to confirm the transaction with their phone, then it does fulfil the requirements, which will be most situations since the fingerprint readers are abundant on mobile phones; otherwise, it would not fulfil the requirements.

There is research from the PEC University of Technology, which found nine different threats with fingerprint sensors [24]. Example of some threats is replay attacks in communication between the fingerprint sensor the database which gets checked and trojan in the system which tricks the system that the fingerprint is a match [24]. What this means in practice is that the fingerprint can be bypassed if the culprit has access to the device.

4.1.1.2 Face Recognition

When it comes to the central bank of Europe's criteria, Face recognition authentication provides the (iii) criteria of being biometric. If the implementation of Face recognition requires the user to confirm transactions with their phone, it fulfils the second criteria of the central bank of Europe [23] because the user is dependent on something it owns (the mobile phone).

The security of face recognition is hardware where it differs based on the technology used if it's 2d or 3d face recognition. There have been cases where phones been tricked with pictures which is not good for something which stores a key.

4.1.1.3 Passcode/Password

Passcode and password authentication fulfil the criteria (i) & (ii) from the European central bank [23]. The security of passcode or password depends on the combination the user chose. Depending on

how randomness and length a password can differ in security. Pin-code consists of numbers, and their length is usually four, which isn't the most secure compared to a password [25]. The possibility for a brute force is slight where the mobile phones block a user who tries wrong pin/passwords multiple times. If the user has a difficult pass/pin and their mobile-phone blocks to many tries, it will be hard for someone to brute-force. The biggest risk is therefore that someone come over the pin code/password or that the user has chosen a too easy code.

4.1.2 Firebase/Google

Authentication for Firebase is via Google, where the Security can be good depending on the user's preference. The user can easily turn on the two-step verification where each time the user logs in, they get a verification code through an SMS-message or a Google-app notification to their mobile-phone [26]. To be able to authenticate themselves, they must, therefore, have their email, password and mobile-phone available. When using one-step verification, the user only needs to have email and password, and the downside is that if someone gets the users information, that person can authenticate themselves like that user [27]. Google has had some security flaws where Google last time got flaws via their old social network Google+ [28]. Therefore, it can be good for a user to have two-step verification for authenticating themselves instead of one-step verification, which is vulnerable.

Google Authentication is Convenient where users with one-step verification can authenticate themselves on every mobile phone. This makes it very easy for the user to access their account if they have forgotten their mobile phone. Although if the user chooses to have two-step verification, they need to have their mobile phone with themselves for being able to authenticate themselves because the verification is device-specific. The downside with both of the verification methods is that the user can forget their password and have issues authenticating themselves.

When it comes to the central bank of Europe's criteria by using only one-step verification, it only satisfies the criteria (i) by letting the user sign in with a password/code. For satisfying the recommendations, however, just having one-step verification isn't enough because it only satisfies one of the criteria. If the user chooses two-step verification, it will fulfil two of the criteria which is (i) for password and (ii) for needing an extra device like a mobile phone for verification. Usage with two-step verification does, therefore, fulfil the safety recommendation. However, there is a problem where Google does not offer re-authentication. That means that when a user signs in for the first time, the session will be saved on the mobile device. This will mean that only the first sign in with two-step verification fulfils the recommendations were on the second sign in the user does not have to authenticate themselves, therefore showing no credentials.

4.1.3 Facebook

Authentication for Facebook is like the one from Google, where there are two ways to authenticate either with one-step verification or two-step verification for more security [29]. The difference is that Facebook has a "Code generator" if you are signed in on another place you can verify by getting a code from Facebook's webpage. This makes it possible for the user to authenticate themselves without their phone as long as they have a logged-in session of Facebook available [29]. Facebook has had some leaks in recent years, which is more than what Google has had. In 2018 there were two attacks where one leaked 6,8 million pictures [30], and the other leaked 30 million Facebook account [31]. They are worrying numbers, and if Facebook is used for authentication, two-step verification is recommended.

When it comes to the central bank of Europe's criteria, it is like the one mentioned in Google/Firebase implementation where one-step does not fulfil the criteria while two-step does. Facebook does not offer re-authentication like Google, and therefore, it's only safe with two-step verification the first sign-in like the Google authentication.

4.1.4 BankID

BankID is a Swedish service for authentication users like an identification card. BankID follows two of the criteria from the recommendations of the European Central Bank. The criteria (i), something only the users know and criteria(ii) which is something the user owns like their mobile phone. It has high security, where twelve banks collaborate to hold up the service [32]. For the users to being able to have BankID, they have to have one of the banks that issue BankIDs. The application itself has not have had any security problem, but there is a loophole where people have been able to trick other people with different methods. One popular method perpetrators use to bring out information about a specific person, call the person and claim that they are from the bank [33], [34]. Then they tell the user to sign for authenticating to the bank, but in reality, the perpetrator is going to sign into that person's account, and very little information is provided on screen. This information on the BankID may even convince the person that they are talking with on the phone is really from the bank [35]. Instead, it results in that someone else gets access to their account.

BankID has implemented a solution to encounter those problems with the help of QR-codes. Instead of just inserting the personal identity number and signing in, the user also has to scan a QR code. This QR-code is a second security measure to make sure the user knows what they are authentication themselves for [36]. The QR-codes have, however, not been implemented everywhere yet, which could be a security risk for some users where their keys can get into the wrong hands.

There are several steps where BankID can fail on the users, either the user forgets their pin-code, fingerprints get broken, or the front-facing camera with faceID fails. If this happens, the user can easily contact the bank and recover new BankID with their real identification card.

4.2 Storage Evaluation

In this section, we will evaluate the storage methods based on the criteria listed in 3.2. This evaluation will be described in the subsequent sections and will also be summarised in table 1.

4.2.1 Offline Storage

This method does not require any connection to a server, and therefore, the key cannot be intercepted in that way. One Security risk with having the keys on paper is that if someone sees or take a photo of the paper, the user will risk the keys [15]. Such risk does not exist, for example, with cash if someone takes a picture or sees a code on the money. Regardless of whether the key is stored on paper or a drive, it must be kept safe. This does, however, require additional measures such as being held in a safe or somewhere secure. When losing the paper, the key will also be is lost, and there is no possible way to recover the key. The only way to be safe from this is to have a copy of the paper, but if someone finds it or the original, the funds are still at risk.

4.2.2 Local Storage Device

A local Storage Device that is connected to an online computer has some security flaws where, for example, unauthorised software and malware easily could track the keys and try to take out important information about them. If this happens, the account holder is at risk of losing everything. The device being online makes it possible to outside threats to send out information that later uses this to take the money. For Local Storage Devices which are offline most of the time this kind of security flaws don't exist, (Resistant to physical observation) the only flaw is if the user does not take good care of the placement of the storage device, someone could plug it in and steal the information(Nonresistant to Physical theft). The user should also be careful where the device is being used because if it's being used in an unsafe environment, the risk of being exposed to information-stealing malware is high [15].

The secure key storage provides better malware resistance than the regular local storage since it does store the keys on a memory space separate from the operating system and is, therefore, said to be malware resistant. The keys cannot be said to be offline since they are on a device that has internet access, but the keys are also not directly accessible from the internet, so it is on a middle ground between online and offline. On top of these security aspects, the secure key storage also provides great convenience as the keys are stored on the device and only requires authentication to be accessed.

4.2.3 Hosted Wallets

On Firebase, Google has set some very strict rules on how the data is managed, and the data is encrypted and cannot be seen by other users. Some small amount of employees can, however, have access to personal data [37], but for accessing this, they must have to have good reasons, which is still troubling when the content is as important as blockchain keys. Google does keep logs of its employee's activities, but it is still something that has to be taken into consideration, but on another hand, it's impossible to reverse blockchain actions when a transaction has been done [37]. Since Firebase would be considered a Hot hosted wallet, we can see in figure 1 that it provides some of the weakest security among the alternatives. Firebase is, however, one of the most convenient alternatives since the keys are always available, but this comes at a risk. On the one hand, there is the risk of a Google employee getting access and being able to observe the keys, and on the other hand, there is also a constant risk of physical theft to a third-party since the keys are always stored online.

Dropbox files are encrypted with 256-bit AES, and it uses SSL to transmit data [38]. The data that is owned by the customer is private and can only be seen by the customer and nobody else [38]. There have, however, been some security leaks passed the years with Dropbox where 68 million account information was sold on the Darkweb [39]. Other than that Dropbox largely suffers from the same issues as Firebase in the sense that they would both be considered Hot wallets.

Neither of these hot storage solutions is ideal from a security point of view, but they do provide very convenience since the keys are always available. In direct opposition to the hot storage is the cold hosted wallet that provides much better security but very bad convenience, to the point where they would be hard to use [10]. The hybrid, being a combination of these two technologies, can provide better security than the hot storage and better convenience than the cold storage. Exactly how secure the hybrid storage is, does, however, depend on the implementation so this alternative becomes hard to evaluate [10].

4.2.4 Security Token

One way to store keys on local storage in a safe way is to use hardware called a Security Token. One example of an existing device that has implemented this type of hardware is Trezor [40]. There is a difference between having the key on Local storage/paper than on a security token like Trezor [41]. The difference is the security which comes with using security tokens like Trezor. There are a lot more security aspects with using Security Token, where Trezor could prevent dangerous situations like when a computer gets malware and viruses. There is also a protection to a fake transaction where an untrusted computer may want to trick the user during a transaction. The Security tokens have a verify function where the user could see on the device where the money is really going to. The security token also has security against theft and being lost if a thief takes the security token they can't access the keys just by observing the token. They need to have the code for getting access to the keys.

Category →	**Local Storage**	**Offline Storage**	Hosted Wallet (**HOT**)	Hosted Wallet (**COLD**)	Hosted Wallet(Hybrid)	Secure Key Storage	Online Banking
Example	Internal Hard Drive	Paper, External Hard Drive, USB Flash Drive	Storing money in our normal Bank Account Service	Storing money in a Savings Bank Account without the ability to take out money frequently.	Storing money in a Savings Bank Account with availability to take out money now and then (more frequently than the cold wallet)	Android Key Storage, Secure Enclave(Apple)	Swedbank, SEB
Malware Resistance	Middle	Middle	No	Middle	No	Yes	No
Key(s) Kept Offline	Middle	Yes	No	Yes	Middle	Middle	No
No Trusted Third Party	Yes	Yes	No	Yes	Middle	Middle	No
Resistant to Physical Theft	No	No	No	No	No	No	No
Resistant to Physical Observation	No	No	No	No	No	No	No
Resilient to Password Loss	Yes	Yes	Yes	Yes	Yes	No	Yes
Immediate Access to funds	Yes	Middle	Yes	No	Yes	Yes	Yes
No New User Software	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cross-device Portability	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 1, showing all the storage alternatives and whether they fulfil the storage criteria introduced in chapter 3 or not [15].

4.3 Storage Of Two Keys Comparison

In multi-key solutions, an interesting scenario is when keys are stored in the same location, it could prove very bad if that location was compromised. Therefore, it is necessary to look at each of the storage methods to see how well they work together.

4.3.1 Paper - Paper

Advantages of having both on Paper is the security which comes with storing the keys offline. The problem is that if the two keys are not kept separately, they can be lost together, which is bad. Another problem is that the keys have a high risk of physical observation where if someone observes them both, then they can easily authenticate themselves [15].

4.3.2 Paper - Local Storage

If one key is stored on paper and one on Offline Storage, the problem that occurs is like having both on paper. They can be easy to lose, and if someone observes the keys, they are at risk. If one is local storage (connected to the internet) and one on paper, the risk of observation increases for the key because the device is online [15]. If the local storage has a KeyStore system, the keys will have increased security, although the device is connected online [20], [21]. The only benefit with this method is the convenience where the user only has to carry one of the keys but pays for it with less security since the risk of losing the key on local storage is higher compared to on paper.

4.3.3 Paper - Firebase

In this case, one key on paper and one on Firebase is a safer alternative than having one key on Local storage, which has internet access [15]. There are, of course, some limitations that Firebase could result in the unauthorized observation of the keys, but the chances are much lower than having it on Online Local Storage [37]. This alternative is decent if the Firebase is authorized with two-step login, where the risk of observation is as low as possible [27]. If the key gets leaked, the observers can't do much because the other key is on paper.

4.3.4 Paper - Dropbox

This alternative is like the Google one except that Dropbox hasn't explained its policy in the same way as Google [38], [42]. Where Google track their employees and that they have to do a two-step verification to access peoples data [37], [42]. Therefore it could be a bigger risk using Dropbox where such strict policy isn't implemented.

4.3.5 Paper - Security Token

Advantages of having one key on paper and one on a Security Token is the security. None of the keys are online and therefore are not at risk of being observed by malware. The problem with having both offline is the risk of losing the device or paper. If the paper is lost or observed, another can quickly get the key, but when having one on security token if lost only the key is lost, nobody can take the money. There is a possibility to get back the key with the help of security token where the user

has a recovery seed which can be saved on paper or somewhere else that is secure [22]. However, if that seed is also lost, there is no hope for that key.

4.3.6 Local Storage - Local Storage

Having both of the keys on Local Storage, which is offline is like having it the keys on two papers. But having them on a device which is connected to the Internet is a big risk. Software and different unauthorized malware can easily, without the user knowing, take key information and use them. This is the worst alternative for storing the two keys [15]. This problem does not apply to the Key Storage system where different encryption techniques are used for securing the keys from other application/malwares [20], [21].

4.3.7 Local Storage - Firebase / Dropbox

One key on local storage offline and one on Firebase is a good alternative since this alternative could be compared to having one key on paper and one on Firebase. If the user has one key on local storage which has a connection to the internet and one on firebase the alternative isn't good because the Local Storage has a high potential of getting observed of software and malware. It's still safe in perspective of not losing everything, but the user has already lost one key, which means that if Firebase security fails, there is no hope for the user. To be able to have the key to local storage and being online Key Storage System can be used to avoid the risk of losing the keys [20], [21].

4.3.8 Local Storage - Security Token

Having one key on offline local storage and one on a security token is like using the same technology but one without security. The risk of losing the devices is high because the devices are small and could easily be stolen. If the devices are stolen the key on Local storage will be lost because the thief can use the key, but on the security token, the thief can't do anything without a pin-code. If the local storage is connected to an online device most of the times, we could count in that the key has already been taken because of the security risks of a key being online. This problem does not apply to the Key Storage system where different encryption techniques are used for securing the keys from other application/malwares [20], [21].

4.3.9 Firebase - Firebase or Dropbox - Dropbox

Having two keys online and at the same service is a big risk. If there is a potential attack or if someone observes the key, both of the keys could be gone.

4.3.10 Security Token - Firebase

The difference between this alternative and Paper - Firebase is that this method implements a more secure method of “offline storage” of the key which makes it less possible to be easily stolen by just observing the key [15]. The only problem is that if the user loses the token, pin and recovery seed, they will lose their key, but it still has recovery methods which the paper does not if the paper is lost there is no way to get access to the key again.

4.3.11 Firebase - Dropbox

A better alternative than Firebase - Firebase or Dropbox - Dropbox because the keys are stored at different places online, which will decrease the risk of losing both of the keys. There is, however, a larger risk that comes with storing both the keys online since both of them could potentially be compromised.

4.3.12 Security Token - Security Token

Security aspects are high, but the problem is that the risk of losing the devices could also be high. A normal user may keep both of the security tokens nearby, meaning that if the tokens are lost or stolen, there is a risk that they are lost or stolen together, which is bad for the user.

4.3.13 Hybrid Storage - Paper

Both paper and Hybrid Storage have high security individually, which makes this a secure alternative. The only downside is that both are not online all the time and requires that you load the key into the application each time you need it.

4.3.14 Hybrid Storage - Local Storage

Hybrid Storage has high-security aspects where local storage depends on the type of that is used. If the Local Storage is insecure, we can count on that one of our keys are already lost. In that case, Hybrid Storage will be our only safe-way where one of our keys already have been observed. On the other hand, if the Local Storage is offline, it will be like Hybrid Storage - Paper. If the Local Storage uses Android Keystore or Secure Enclave, the keys are stored securely. The advantage of this compared to Local Storage offline is that the Keystore systems can be accessed easily by the user and be safe at the same time. Making this alternative more reliable for the user.

4.3.15 Hybrid Storage - Hot Storage

The advantage of this alternative is that both of them are accessible at any time and are not bound to a specific device, which is very convenient. The downside of this is that only one of the keys is stored securely while the other one is at risk.

4.3.16 Hybrid Storage - Hybrid Storage

While hybrid storage is one of the more secure alternatives, there are still risks involved with it. So to put both keys on hybrid storage is not ideal as if one key gets compromised, it is likely the other one is at risk too. This approach is, however, convenient for the user and not bound to a specific device.

4.3.17 Hybrid Storage - Security Token

This alternative is similar to 4.3.13 in terms of security but with the key difference that this alternative is not susceptible to physical observation with the paper storage

Storage	Paper	Local Storage	Hot Storage	Hybrid Storage	Security Token
Paper	**Risk of losing both the papers.**	**Risk when one key has internet access.**	**Risky having key online**	**Good Security**	**Good Security**
Local Storage	**Risk when one key has internet access.**	**Both keys on a Storage with Internet access.**	**Both keys on a Storage with Internet access.**	**Risky having key online.**	**Risk when one key has internet access.**
Hot Storage	**Risky having key online**	**Both keys on a Storage with Internet access.**	**Both keys online on the same database.**	**Risky having key online.**	**Risky having key online.**
Secure Key Storage	**Good Security**	**Risk when one key has internet access.**	**Risky when one key has is always online on third-party service.**	**Good Security**	**Good Security**
Hybrid Storage	**Good Security**	**Risky having key online.**	**Risky having key online.**	**Single point of failure**	**Good Security**
Security Token	**Good Security**	**If storage with internet access**	**Risky having key online.**	**Good Security**	**Risk of losing both the devices.**

Figure 2 shows different security risks by storing two keys in different ways.

RED – Should be avoided has a high risk of keys being lost/stolen
YELLOW – Key storage which some risks of being lost/stolen
GREEN – Key storage which small or none risk of being lost/stolen.

4.4 Recovery

This section will be an analysis of the recovery techniques introduced in chapter 2 and the credentials needed to perform a recovery.

4.4.1 Two Signing Keys, One Recovery Key

One alternative is to have three blockchain keys but to use one as a dedicated recovery key, which means the other two keys become signing keys. So, in this case, the recovery key is the necessary credential for a Recovery. With the one-key signature solution, if the key is lost, everything is lost, and the user has no way to get back the funds connected to that key. However, with multikey solutions with a recovery key, the user can lose one signing key and still being able to get access to their funds. Storage of the recovery key can be very similar to the storage of the regular keys as described previously. The difference here is that the recovery key is not required when making transactions, which opens up possibilities that would have been too inconvenient for the regular key storage. This

does, however, beg the question, what happens if the recovery key is lost? Well, if the recovery key is lost, then the user will not be able to perform a recovery in the future, which is not ideal. Therefore, it might be interesting to look at some other multikey techniques.

4.4.2 Multikey Recovery Options

When having dedicated recovery keys, the possibilities when losing keys depends largely on how many keys are necessary for signing and recovery. Hence, in this section, we examine what possibilities exist upon losing keys. In the previous section, the possibilities with one separate and dedicated recovery key, in this section that idea is expanded upon by examining multiple dedicated recovery keys as well as keys that are both recovery and signing keys at the same time.

First, we look at the scenario where we have separate signing and recovery keys, and one recovery key is lost. Since the singing and recovery keys are different, the number of keys needed for a sign is not relevant, and instead, we can only focus on the number of keys needed for recovery. The table below shows these scenarios with the “M of N” requirement for each row, where M is the required number of keys, and N is the total number of keys connected to that action.

Recovery M of N	**Stolen**	**Lost**
Recovery 3 of 3	The thief cannot recover, but the user can.	The user can no longer recover a signing key.
Recovery 2 of 3	The thief can’t access account if only one key has been stolen. If only one key is stolen, a recovery can easily be made by the user with the help of the two other keys.	There is no problem if one key is lost because recovery can easily be made with the two other keys. If one more recovery key is lost, then the user will not be able to recover any more signing keys.
Recovery 2 of 2	The thief cannot recover, but the user can.	The user can no longer recover a signing key.
Recovery 1 of 3	The perpetrator can now recover one of the users signing keys.	The user can now still replace a lost or stolen signing key as long as they do not lose the rest recovery keys.
Recovery 1 of 2	The perpetrator can now recover one of the users signing keys.	The user can now still replace a lost or stolen signing key as long as they do not lose another recovery key.

Recovery 1 of 1	If the recovery key is stolen, there is no possibility for the user to recover any key. This also gives the possibility for the perpetrator to recover one of the users signing keys.	If the recovery key is lost, there is no possibility for the user to recover signing keys which may be stolen/lost in future. The user still has the availability to sign and access their account.

RED – Bad scenario where the keys are either in risk or where user can no longer use their keys.
YELLOW – OK/Worrying scenario where the keys can be in risk or that the user has lost their possibilities to recover their keys.

We can see that none of these scenarios is particularly good, and even when the funds are not necessarily at risk, there is one major problem with this approach. The problem is, how do you recover or replace recovery keys? Should we have recovery keys to recover recovery keys? The idea becomes unreasonable fast; hence, this approach is not flawless. The upside, however, is that even if several signing keys are lost, they can be replaced, but if more than one recovery key is lost, then recovery becomes impossible. We can also see that the best alternative out of these is recovery 2 of 3.

Next, we will look at scenarios when all the keys can be used as both signing keys and recovery keys. Since one key can be used in both the recovery and signing process, it is relevant to also talk about the number of keys required for signing a transaction. In the leftmost column, the bold text in the table states how many keys are needed for recovery while the smaller texts state how many keys are needed for signing.

Recovery 3 of 3	**Stolen**	**Lost**
Signing 3 of 3	**Impossible -** Three keys needed for signing; therefore, either the user can sign or the thief.	**Impossible -** Two keys needed for signing; therefore, the user has no access to their account.
Signing 2 of 3	**Impossible -** Two keys needed for signing; therefore, the thief can't sign.	**Impossible / Worrying -** The user has no possibility to recover the second key but can still access the account.
Signing 1 of 3	**Impossible / BAD -** The user has no possibility to recover their second key, and the thief has the possibility to access the users' account.	**Impossible / Worrying** - The user has no possibility to recover the second key but can still access the account.

RED – Bad scenario where the keys are either in risk or where user can no longer use their keys.

Recovery 2 of 3	Stolen	Lost
Signing 2 of 3	**OK / Worrying -** It's still possible for the user to sign as there are only 2 keys needed for signing. The key that was stolen can be replaced through the recovery process, but if the perpetrator gets one more key, then all the funds the user had are no longer protected.	**OK -** It's still possible for the user to sign as there are only 2 keys needed for signing. No keys have been stolen, and the key which has been lost can easily be recovered.
Signing 1 of 3	**BAD -** If one of the three keys gets stolen, the thief can easily sign and get access to the users' money.	**OK -** If there is no physical observation of the key that was lost, then it does not matter much since the key can easily still be recovered with the remaining keys.

RED – Bad scenario where the keys are either in risk or where user can no longer use their keys.
YELLOW – OK/Worrying scenario where the keys can be in risk or that the user has lost their possibilities to recover their keys.
GREEN – Safe where the user has not lost any security aspects or risking their keys.

Recovery 2 of 2	Stolen	Lost
Signing 2 of 2	The thief cannot recover or sign but the user can	Two keys needed for signing, therefore, the user has no access to their account.
Signing 1 of 2	The user has no possibility to recover their second key, and the thief has the possibility to access the users' account.	The user can still sign but cannot recover

RED – Bad scenario where the keys are either in risk or where user can no longer use their keys.
YELLOW – OK/Worrying scenario where the keys can be in risk or that the user has lost their possibilities to recover their keys.

Recovery 1 of 2	Stolen	Lost
Signing 2 of 2	If one of the keys gets stolen, the thief can easily sign and get	If there is no physical observation of the key that was

	access to the users' money.	lost, then it does not matter much since the key can easily still be recovered with the remaining key.
Signing 1 of 2	If one of the keys gets stolen, the thief can easily sign and get access to the users' money.	If there is no physical observation of the key that was lost, then it does not matter much since the key can easily still be recovered with the remaining key.

RED – Bad scenario where the keys are either in risk or where user can no longer use their keys.
GREEN – Safe where the user has not lost any security aspects or risking their keys.

We can see that recovery with 2 of 3 and sign with 2 of 3 is the best option. This implementation is, however, also not perfect as it means that a user that cannot lose more than one key without also losing their funds. The advantage is, however, that only three keys are needed instead of the six keys in the previous method.

4.4.3 Biometric Encryption

The point of encrypting the keys this way with biometrics rather than with passwords is that is for avoiding physical observation, but there are some problems with doing that. By using passwords, it is possible to easy hash and add salt to make it hard for someone to observe the keys. Biometrics is on the other side is not a good alternative to use for hash because each biometric could create a different hash. For example, how the finger/face have been placed on the biometric sensor and very small changes will make a difference in the hash [43]. If the hash changes, there is no possibility to identify that the same finger/face are being read. Therefore a possibility to hash and salt and encrypt keys with biometric is not a good choice, and there is also security flaws that come with it [43]. It is easy to find a fingerprint of a person but harder to figure out a password. If someone loses their phone, a thief can easily take the fingerprints which are on the phone, copy them and then trick the reader that they are the real user [49]. Problem with a fingerprint or face recognition is that if someone finds the hash or copy, there is no way to reset it, but a password can be changed [43].

4.4.4 Recovery Seed

An alternative to the idea of recovery keys is the recovery seed, which is used in e-wallets like Trezor [40]. As mentioned in chapter 2, the recovery seed can restore an entire e-wallet, which is an advantage it has over all the other alternatives mentioned, but this comes at a price. If the seed is physically observed or otherwise accessed by someone other than the user, they can restore the user's entire e-wallet and use all the funds. The Recovery Seed, therefore, has to be kept in a secure place where nobody else can get access to it [15]. The recovery seed is a long-time safety for the user where the seed can be used in a long time [22]. This implementation can be used in both software e-wallets and hardware e-wallets.
If the recovery seed is lost, there are no possibilities to recover back the blockchain wallet again.

There are different methods to keep the recovery seed secure; an example is Crypto steel. Cryptosteel is a little tool where engraved letters/numbers can be kept in and also be locked. It's also

secured from falling from high altitudes and fire [44]. If the recovery seed had been on a paper, the recovery seed would burn away, and the user loses their wallet.

4.4.5 Third Party

The advantage of storing a dedicated recovery key at a third party is that the recovery key cannot be accessed as easily by a perpetrator. One problem that exists, though, is that more and more safety deposit boxes are closing, which makes them less accessible [45]. This is an option that was not possible with the signing process as it would be too inconvenient fetch the key from a bank or have a friend confirm the transaction. But, especially with the case of dedicated recovery keys and recovery seeds, this inconvenience factor is almost negligent since they are not meant to be used frequently.

5 Design, Implementation and Evaluation

In this section, we will discuss the proof of concept application that was made in this project and motivate the implementation choices that were made along the way with respect to the research that was done in chapter 4.

5.1 Implementation

First, we must choose the recovery method since the number of keys decides how many storage options are necessary. The best alternative for recovery, as stated in the previous chapter, is the recovery seed but implementing the recovery seed would not be very valuable as a proof of concept since it already exists in many e-wallets. Instead, we chose to implement the setup with three keys that can both be used as sign keys and recovery keys. We could also have chosen to take the separate signing and recovery keys approach, but this would simply require more storage options and would not necessarily prove anything that the other approach could not. Both the sign and the recover should require two out of three keys.

For the storage options, we know that we need at least three of them since we have three blockchain keys in our implementation. For storage, the best alternatives are security token and hybrid storage, while these are the most secure options they are also the most advanced and therefore could not fit within the timeframe of this project. This leaves Secure key storage, offline storage and hot storage. Secure key storage and offline storage are not bad alternatives by any means and do provide good security, but hot storage such as with Firebase and Dropbox is one of the least secure alternatives. However, since it was concluded in the previous chapter that storing multiple keys on the same platform is also not ideal hot storage with Firebase was implemented. While not the most secure alternative, the Firebase storage might be interesting from a proof of concept perspective since it would prove that signing and recovery are possible without having the key be stored or loaded into the mobile device.

As for authentication, it was not possible to implement BankID into the application because a contract is necessary with one of the banks that are in charge of BankID [46], [47]. Local authentication was a good alternative and is not very hard to implement either, so this option was included in the proof of concept. Finally, since Firebase was chosen as a storage method, there was a need for an online authentication method. This is because we could not use local authentication with password or fingerprint to log into Firebase, which is why Facebook and Google login was implemented even despite the many issues they have.

The storage and authentication methods used together are:

- **Paper Storage with QR-code**
  Paper storage with QR-code means the user needs to scan the QR-code into the application every time the user wants to sign a transaction or recover a key. Authentication for this type of offline storage is simply having the paper/QR code, as that counts as a credential. Disadvantages with this storage are that keys can only be accessed when the paper/QR-code is nearby.

- **Android Key Storage/ Secure Enclave with Local authentication**
  Secure key storage with local authentication means that the user needs to authenticate themselves with their password, PIN-code, fingerprint, or whatever it might be in order for the key to be decrypted from the secure storage. There are some risks with the fingerprint which

we have mentioned earlier, but despite those reasons, popular banking systems use fingerprint scanning, and therefore, the risks might be at an acceptable level [14].

- **Firebase Storage(Hot Storage) with Google and Facebook login**
  With Firebase storage, there are two ways of handling the key. Either the key can be stored on Firebase and fetched to the application each time it is needed, this is the easy alternative. The other alternative is to have the key stored on Firebase but also never have it leave there; instead, it is possible to write Javascript scripts that use the key to make a transaction and then send the transaction to the application or directly to the blockchain. Authentication for accessing storage is either by Firebase own username & password login or Facebook/Google login. It should be noted that out of the three storage and authentication pairs, this is the most unsafe one as it combines the most unsafe storage option with the most unsafe authentication methods.

5.2 Design

The user starts by signing up for a new account or logging in to an existing account. If the user is signing up for the first time the user has to place their keys on three different storage places. Where in the proof of concept, there are three alternatives available: offline storage(QR-Code), Firebase Storage and Internal Storage(Secure enclave/Android Key storage).

The login process consists only of entering a username which is not stored in the app but rather registered on the blockchain. After entering the username, the user gets to choose whether they want to sign a transaction or recover a key. Both cases end up with the user being presented with the three storage methods, each with their authentication method as well.

The idea is that the user should authenticate themselves on enough of the methods to get the keys necessary for a sign or a recovery. The difference between sign and recovery is that with recovery, after authenticating two of the alternatives, the user should be able to replace one key that will be generated anew.

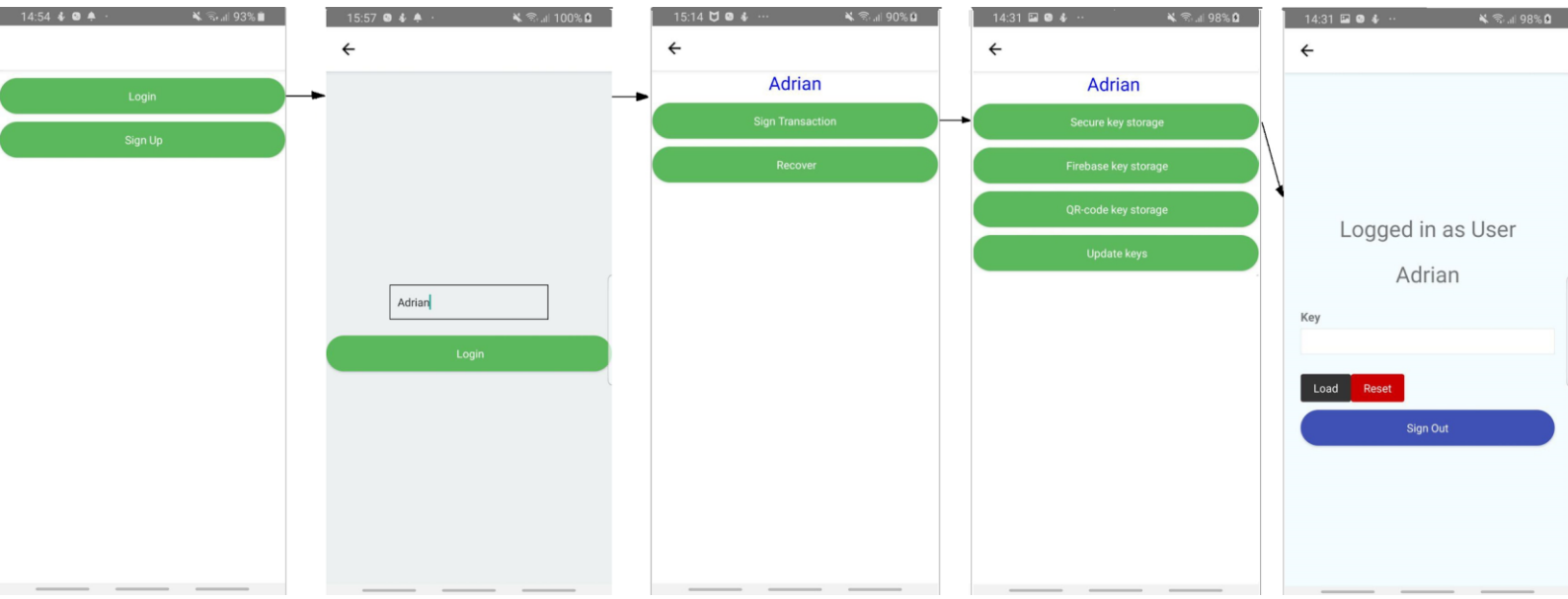

Figure 3. Illustration of a user loading their saved keys from Secure key storage

Figure 3 is an illustration of how Secure key storage for loading their keys can be used. The user starts with pressing login, then gets redirected to the next page where they have to enter their username. When they put in their username, they have to press Login and then be redirected to a personal page. In the personal page, their username will be presented at the top, and they have two choices to make. The user can either sign transaction or recover keys. If the user chooses to sign transactions, they will be redirected to the next page where they have to choose where they have stored their keys to be able to access them. In this case, the user has saved one key on Secure key storage. By pressing on Secure key storage, the user will be redirected to a page where the user either reset their saved key or access it with the load button.

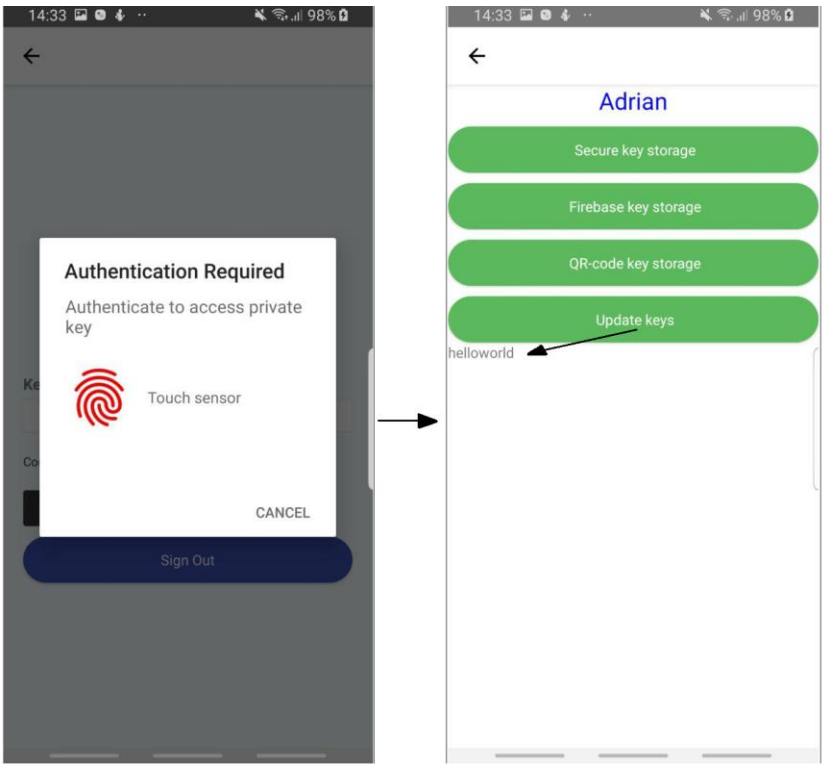

Figure 4. Illustration of how the user has to authenticate itself for getting access to their keys.

If the user wants to load their key, the user has to authenticate him/herself with either fingerprint or Pincode/password depending on what is used for local authentication. When the key is loaded, the user can use update keys to access their keys. Figure 4 has illustrated how the authentication can be made by fingerprint and how the user can use the update keys button.

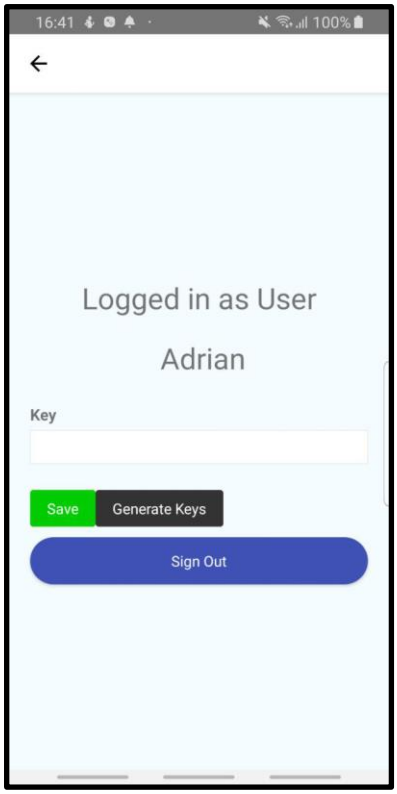

Figure 5. Illustration on how Secure Key Storage looks when configuring a new user.

If the user has signed up instead of logging in the procedure will be similar but with some small changes. The changes in Secure Key Storage will be that instead of loading a key in the signing up process the user has to save their keys so when they later login are able to load and use them for signing transactions. Figure 5 illustrates the page where to save and generate keys exists.

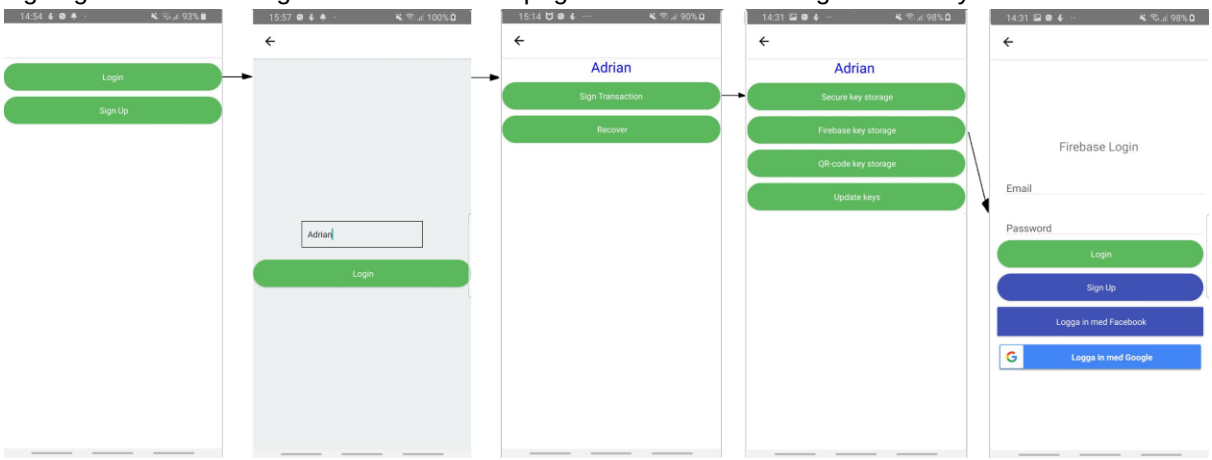

Figure 6. Illustration of how a user loading their saved keys from Firebase

For loading keys from Firebase, the procedure is similar until the page where the user chooses their storage alternative. When the users choose Firebase, they will be redirected to an authentication page where the user has three different authentication alternatives. Figure 6 illustrates how to get to the authentication page and the different authentication methods that exist.

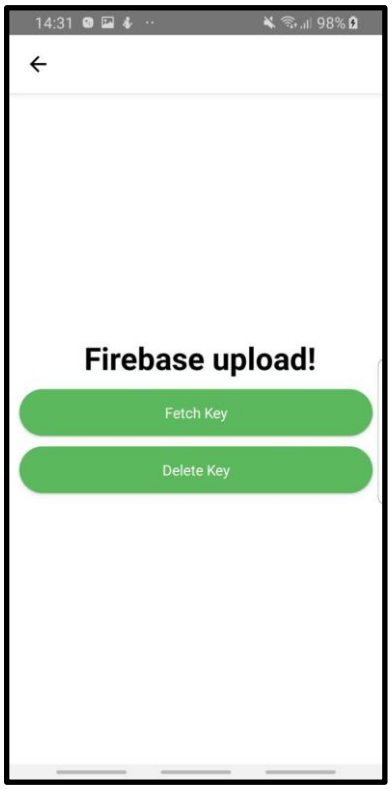

Figure 7. Firebase load key page

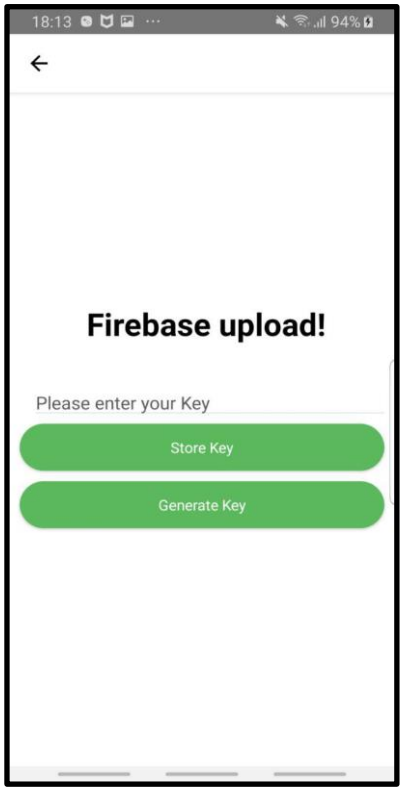

Figure 8. Firebase Storing key page

Figure 7 illustrates when a user has authenticated itself on Firebase and gets redirected to the page where the user can either load or delete their key. If the user is signing up for the first time instead of going to the loading firebase page, they will be redirected to the firebase page illustrated in figure 8 where they can store key.

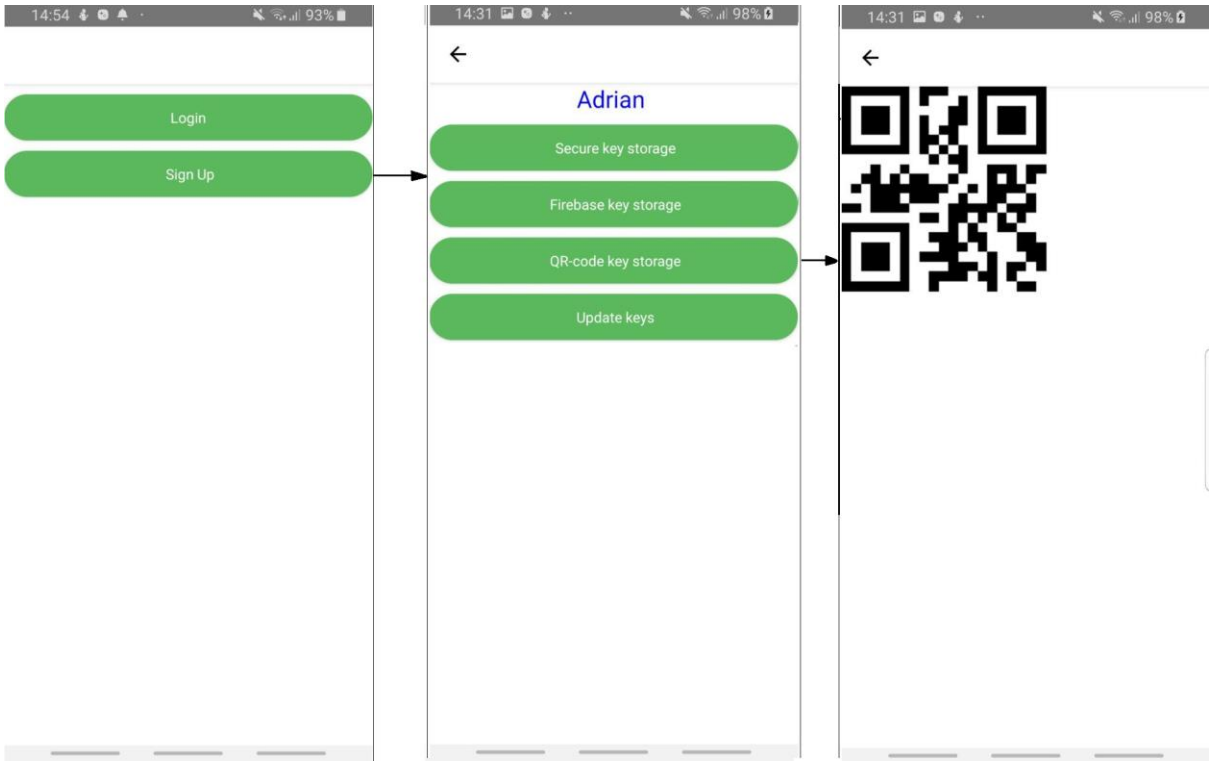

Figure 9. *Illustration of how a user saves their keys on a QR*

When the user is signing up, they have the choice to generate their key on a QR-code. By pressing on the function QR-code key storage, they will get a QR-code which represents their key. Figure 9 illustrates how that is done and how it looks like when the user gets their key on a QR.

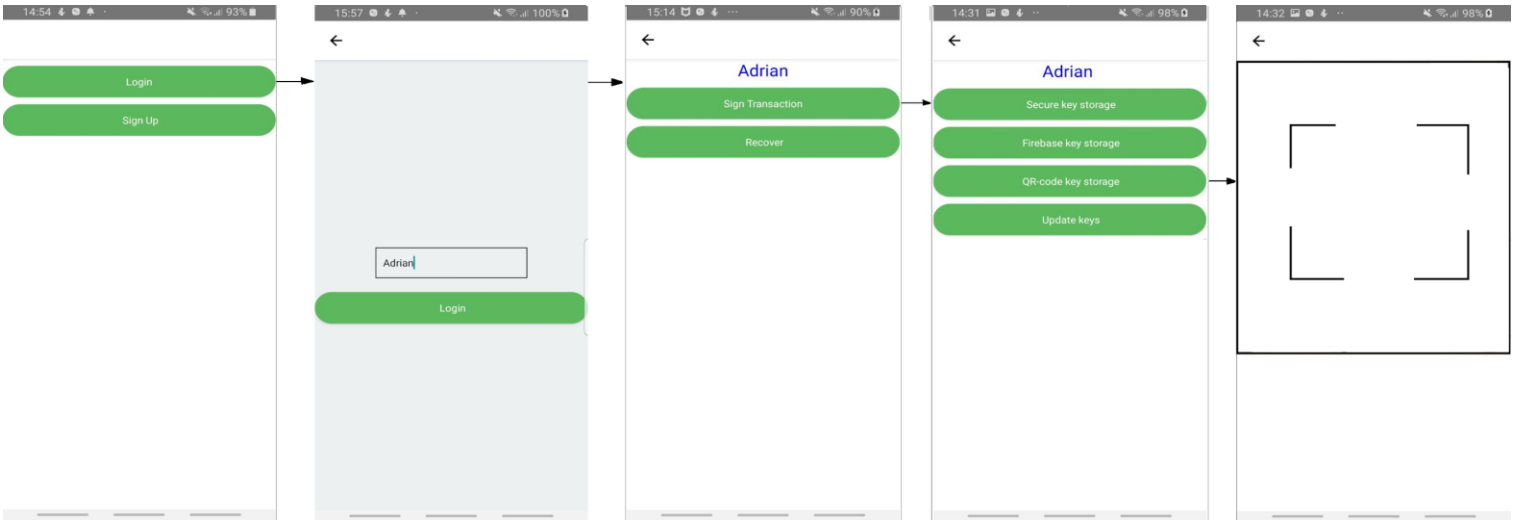

Figure 10. Illustration of how a user loads their keys from a QR

When the user is done signing up, they can later load their key by logging in and then by pressing the same QR-Code key storage button they will get redirected to a scanning function where they can access their key. Figure 10 illustrates the procedure the user has to go through for scanning/load their key.

5.3 Evaluation

Due to the limitations in the project, it has become apparent that the proof of concept solution is not ideal and therefore it is interesting to evaluate it based on the criteria defined in chapter 3 to see exactly how it holds up.

First, we look at the storage. The solutions that were used here are paper storage, Firebase and secure key storage. For each of the storage criteria, we will evaluate how well the solution handles it.

- **Malware Resistance** - Firebase, being a hot wallet, cannot be assumed to be malware resistant since hot wallets, in general, are susceptible to malware while paper storage is by nature not susceptible to malware. The secure key storage is an interesting middle ground here since the phone can, in general, definitely be susceptible to malware, but it is unclear how susceptible this storage is. This criterion is, therefore, at best, only partially solved.
- **Keys kept Offline -** Paper storage obviously keeps the keys offline while Firebase obviously does not, the secure key storage is once again in an interesting middle ground here. Keys stored with secure key storage can definitely be on a device connected to the internet but are not accessible through the internet since the access can be bound to the app that stored the key in question. Like the previous criterion, this one is also one partially solved at best.
- **No Trusted third-party -** When it comes to paper storage, there is not a third-party, but on secure key storage and Firebase this is not as simple. With secure key storage, a certain amount of trust is put in Android/IOS to store the keys, but it is not possible for them to say, sign a transaction so this may not be a huge issue. With Firebase, however, the keys are

more or less completely in Google's control, which is bad. This problem is mostly solved since two out of three methods solve it.

- **Resistant to physical theft -** None of these solutions stores the keys in a way that is resistant to physical theft, this problem is therefore not at all solved
- **Resistant to physical observation -** the same as the previous criterion.
- **Resilient to Password Loss -** All of the storage methods are resilient to password loss. This is obviously not an issue for paper storage, for secure key storage it does depend a little on what type of local authentication is used since biometric authentication cannot be forgotten while regular passwords and PINs can. With Firebase password loss would not be a problem since the passwords for Firebase authentication can be reset. This problem is, therefore, more or less solved.
- **Immediate Access to funds -** With both Firebase and secure key storage the user would have immediate access to their funds, but this is not the case with paper storage as it has to be scanned into the application. This is a case where convenience is sacrificed for security, so while the criterion is not satisfied, it is intentional.
- **No New User Software -** Other than the application, no additional software is required for any of the solutions; therefore, this criterion is considered solved.
- **Cross-device Portability -** Both Firebase and paper storage are cross-device-portable as they are not bound to any specific device, this is, however, not the case with secure key storage as the key stored there would not be accessible on another unit. The criterion is therefore considered partially solved.

Previously in this report, we have evaluated the authentication methods individually with the ECB's recommendations, but those recommendations are not necessarily for each authentication method but for the transaction as a whole. Since we require two out of three keys to sign a transaction, we need two authentication methods which must together meet the ECB's recommendations. From combinatorics, we know that choosing a subset of two elements from a set of three elements can be done in three ways, all of which will be analysed below.

1. **Secure key storage & Paper storage -** Paper storage only provides one credential since the paper is something the user owns. Secure key storage always needs two credentials, something the user owns with the phone and either something the user knows if the local authentication uses password/PIN or something the user is if biometrics are used. This transaction, therefore, requires at least two credentials so it does meet the recommendations.
2. **Secure key storage & Firebase -** In the previous chapter, we concluded that Facebook/Google login could actually not require any credentials at all if there is a signed in session on the device. But from the previous transaction, we know that secure storage requires two credentials so this transaction would also need the recommendations.
3. **Firebase & Paper storage -** We know that Google/Facebook authentication can require no credentials and paper storage only provides one. This means that this transaction can potentially not meet the ECB's recommendations. The only way to make it meet the recommendations is to log out of any Facebook/Google sessions before every transaction, which is unreasonable, and this is regardless of if two-step verification is used.

6 Conclusions and Future Work

In this chapter, we will look back at the analysis of chapter 4 to conclude which implementations are suitable for blockchain e-wallet and which ones are not.

6.1 Conclusions

We start with the authentication methods and with the Swedish e-identification system BankID, which is deemed to be the best alternative. This is because of its high security, where there have been zero attacks to the system since the release 14 April 2010 [48]. There have only been phishing attacks the recent years where people have been tricked through their mobile-phone to sign transactions claiming they are someone else [33]. These problems are being fixed with a QR-code implementation, which will eventually be implemented in all bank logins [36]. To be able to authenticate, the user has to have their phone, Swedish personal registration number(personnummer) and being able to press their BankID code. Hence, BankID fulfils the recommendations from the European Central Bank [23] concerning authentication and provides some of the best reliability due to the banks backing it which is why it is deemed suitable as an authentication method [46].

Secondly, we have the local authentication by passcode/password, face recognition and fingerprint. All of these techniques fulfil the recommendations from the European Central Bank concerning authentication [23]. There are some risks with all of these techniques where a password can be brute-forced if the implementation does not have a blocking system. A fingerprint also has some risk where the system can be tricked or that fingerprints get copied [7], [43], [49]. There are lots to be said about the specifics of local authentication, but all in all, it does provide pretty good security and is, therefore, deemed suitable.

Lastly, we have Google and Facebook login. These two can fulfil the ECB's recommendations if two-step authentication is enabled, which cannot be enforced. However, even if two-step authentication is enabled, it is still not a certainty that the recommendations are met; this is due to the absence of what Facebook calls "reauthentication". Reauthentication is when the user is forced to sign in with their email and password every time they wish to log in, without reauthentication a previously logged in session is simply used to authenticate the user without email or password [50]. So, if both two-step verification and reauthentication are enabled, then this type of authentication does meet the ECB recommendations [23], but at this time re-authentication is not possible for mobile applications that use the UI provided by Facebook [50]. For Google login, we also could not find anything like reauthentication. This means that Google and Facebook authentication both do not meet the recommendations of the ECB and therefore provides the weakest form of authentication amongst the ones researched in this report and is, therefore, not deemed suitable.

As for storage, we can see in table 1 that secure key storage is the best alternative since it does well in almost all of the criteria that were looked at while also being convenient since it stores everything locally on the phone. We also have offline storage which is the second most secure alternative according to the criteria, if we also store the offline key on paper we get the added benefit of it being malware resistant while sacrificing some convenience [15]. We can also see that Hot wallets like Dropbox or Firebase make for pretty poor alternatives, especially when compared to hybrid storage which can have all the advantages of hot storage convenience but with higher security. The advantage of the online storage is that the key is always available and is not tied to any device, but this can be achieved with hybrid solutions as well, so there does not seem to be any reason to hot storage over hybrid storage from a security perspective. The implementations for both hybrid storage and security token were not brought up in this report, but principally they show promise and are

deemed potentially suitable while paper and secure key storage are deemed suitable and hot/cold storage is not deemed suitable.

For recovery, the most useful alternative is probably the recovery seed as it can recover the whole wallet. The problem with recovery seeds is that they need to be stored somewhere safe such as on paper and if someone else should get the seed, then they can recreate the whole e-wallet and spend all the funds. With all this in mind, it is important to remember that recovery seeds are used in bitcoin e-wallets today, so these risks might be acceptable given the benefits of the seed [22]. As for the recovery key solutions, none of them allows for losing more than one key, but the three signing and recovery keys provide the most security out of the bunch. Biometric encryption was also researched, and while it is an interesting concept, it is unfortunately not usable with today's technology.

6.2 Future Work

Future work could include things that were touched upon in this report but not fully explored, such as hybrid storage and implementations of that. Security tokens are also something that was very interesting due to the safety that it provides. It could be interesting to look at how it could communicate with this type of application. Recovery with the help of a friend is something that also should be researched more about and then implemented in the application. Where with the help of a secure friend, users can provide their recovery key and ask for it every time they have lost one of their keys. Recovery Seeds should also be touched upon on how users could be able to recover their whole e-wallet.

Bibliography

[1] ”Centiglobe”, *Centiglobe*. [Online]. Online at: https://www.centiglobe.com. [Accessed: 14-july-2019].

[2] ”React Native · A framework for building native apps using React”. [Online]. Online at: https://facebook.github.io/react-native/. [Accessed: 14-july-2019].

[3] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, “Bitcoin and Cryptocurrency Technologies,” p. 308.

[4] G. Chen, B. Xu, M. Lu, and N.-S. Chen, “Exploring blockchain technology and its potential applications for education,” *Smart Learn. Environ.*, vol. 5, no. 1, p. 1, Jan. 2018.

[5] Europe, G. Focus, and N. America, “How the Blockchain Will Impact the Financial Sector,” *Knowledge@Wharton*. [Online]. Available: https://knowledge.wharton.upenn.edu/article/blockchain-will-impact-financial-sector/. [Accessed: 07-May-2019].

[6] “Om avancerad säkerhetsteknik med Touch ID,” *Apple Support*. [Online]. Available: https://support.apple.com/sv-se/HT204587. [Accessed: 28-Mar-2019].

[7] V. Goel, “That Fingerprint Sensor on Your Phone Is Not as Safe as You Think,” *The New York Times*, 22-Dec-2017.

[8] S. Singh and S. V. A. V. Prasad, “Techniques and Challenges of Face Recognition: A Critical Review,” *Procedia Comput. Sci.*, vol. 143, pp. 536–543, Jan. 2018.

[9] W. B. Soltana, D. Huang, M. Ardabilian, L. Chen, and C. B. Amar, “Comparison of 2D/3D Features and Their Adaptive Score Level Fusion for 3D Face Recognition,” p. 8.

[10] S. Kovach, “Samsung’s Galaxy S8 facial recognition feature can be fooled with a photo,” *Business Insider*. [Online]. Available: https://www.businessinsider.com/samsung-galaxy-s8-facial-recognition-tricked-with-a-photo-2017-3. [Accessed: 10-May-2019].

[11] B. Gokberk, A. Salah, L. Akarun, R. Etheve, D. Riccio, and J.-L. Dugelay, “3D Face Recognition,” in *Guide to Biometric Reference Systems and Performance Evaluation*, 2009, pp. 263–295.

[12] “Om den avancerade Face ID-tekniken,” *Apple Support*. [Online]. Available: https://support.apple.com/sv-se/HT208[1088]. [Accessed: 10-May-2019].

[13] M. Wemnell, “Statistik BankID – användning och innehav,” p. 9, 2017.

[14] “Mobilt BankID.” [Online]. Available: https://support.bankid.com/sv/fragor-svar/mobilt-bankid. [Accessed: 28-Mar-2019].

[15] S. Eskandari, J. Clark, D. Barrera, and E. Stobert, “A first look at the usability of bitcoin key management,” *Proc. 2015 Workshop Usable Secur.*, 2015.

[16] “Android keystore system,” *Android Developers*. [Online]. Available: https://developer.android.com/training/articles/keystore. [Accessed: 22-Apr-2019].

[17] Apple, “iOS Security iOS 12.1 November 2018,” p. 95, 2018.

[18] T. Cooijmans, J. de Ruiter, and E. Poll, “Analysis of Secure Key Storage Solutions on Android,” in *SPSM@CCS*, 2014.

[19] “Android keystore system | Android Developers.” [Online]. Available:

https://developer.android.com/training/articles/keystore#ExtractionPrevention. [Accessed: 02-May-2019].

[20] T. J. P. M. (Tim) Cooijmans, “Secure Key Storage and Secure Computation in Android.” Radboud University Nijmegan, 30-Jun-2014.

[21] “Storing Keys in the Secure Enclave | Apple Developer Documentation.” [Online]. Available: https://developer.apple.com/documentation/security/certificate_key_and_trust_services/keys/storing_keys_in_the_secure_enclave. [Accessed: 02-May-2019].

[22] “Recovery seed,” *Trezor Wiki*. [Online]. Available: https://wiki.trezor.io/Recovery_seed. [Accessed: 06-May-2019].

[23] “RECOMMENDATIONS FOR THE SECURITY OF MOBILE PAYMENTS DRAFT DOCUMENT FOR PUBLIC CONSULTATION.” European Central Bank, Nov-2013.

[24] M. Kaur, Dr. S. Sofat, and D. Saraswat, “Template and database security in Biometrics systems: A challenging task,” *Int. J. Comput. Appl.*, vol. 4, no. 5, pp. 1–5, Jul. 2010.

[25] B. Frank, “Internet Security,” 18-Jul-2013. [Online]. Available: https://scholar.harvard.edu/files/bfrank/files/internetsecurity.pdf. [Accessed: 10-May-2019].

[26] “Tvåstegsverifiering från Google ‘How It works.’” [Online]. Available: https://www.google.com/landing/2step/#tab=how-it-works. [Accessed: 28-Mar-2019].

[27] “Tvåstegsverifiering från Google ‘How It Protects.’” [Online]. Available: https://www.google.com/landing/2step/#tab=how-it-protects. [Accessed: 28-Mar-2019].

[28] M. Maven, “Google Is Shutting Down Google+ After A Secret Potential Data Leak. Here’s A Profit Lesson Behind It,” *Forbes*. [Online]. Available: https://www.forbes.com/sites/michaelmaven/2018/10/09/google-is-shutting-down-google-after-a-secret-potential-data-leak-heres-a-profit-lesson-behind-it/. [Accessed: 28-Mar-2019].

[29] “Vad är tvåfaktorsautentisering och hur fungerar det? | Facebooks hjälpcenter.” [Online]. Available: https://www.facebook.com/help/148233965247823. [Accessed: 28-Mar-2019].

[30] J. Kastrenakes, “Facebook exposed up to 6.8 million users’ private photos to developers in latest leak,” *The Verge*, 14-Dec-2018. [Online]. Available: https://www.theverge.com/2018/12/14/18140771/facebook-photo-exposure-leak-bug-millions-users-disclosed. [Accessed: 28-Mar-2019].

[31] “Facebook hack leaks data from 30 million users (UPDATE) | TechRadar.” [Online]. Available: https://www.techradar.com/news/facebook-hack-leaks-data-from-50-million-users. [Accessed: 28-Mar-2019].

[32] “Vad är BankID?” [Online]. Available: https://support.bankid.com/sv/bankid/vad-aer-bankid. [Accessed: 28-Mar-2019].

[33] “Varning för falska mejl.” [Online]. Available: https://support.bankid.com/sv/sakerhet/varning-for-falska-mejl. [Accessed: 28-Mar-2019].

[34] “BankID och säkerhet.” [Online]. Available: https://support.bankid.com/sv/sakerhet/bankid-och-sakerhet. [Accessed: 28-Mar-2019].

[35] M. K. / Omni, “Säkerhetsbrister i bank-id – enkelt komma åt konton,” *Svenska Dagbladet*, 30-Nov-2016.

[36] “Varför inför BankID avläsning av QR-kod?” [Online]. Available: https://support.bankid.com/sv/fragor-svar/mobilt-bankid/varfor-infor-bankid-avlasning-av-qr-kod. [Accessed: 28-Mar-2019].

[37] “Privacy and Security in Firebase,” *Firebase*. [Online]. Available: https://firebase.google.com/support/privacy/. [Accessed: 29-Mar-2019].

[38] “Is Dropbox safe to use?” [Online]. Available: https://help.dropbox.com/security/safe-to-use. [Accessed: 28-Mar-2019].

[39] “Dropbox hacked; what you need to know.” [Online]. Available: https://www.kaspersky.com/blog/dropbox-hack/12875/. [Accessed: 28-Mar-2019].

[40] “Security:Security philosophy,” *Trezor Wiki*. [Online]. Available: https://wiki.trezor.io/Security:Security_philosophy. [Accessed: 10-Apr-2019].

[41] “Security:Security philosophy,” *Trezor Wiki*. [Online]. Available: https://wiki.trezor.io/Security:Security_philosophy. [Accessed: 10-Apr-2019].

[42] “Privacy - Google Cloud Help.” [Online]. Available: https://support.google.com/googlecloud/answer/6056650?hl=en. [Accessed: 18-May-2019].

[43] By, “Your Unhashable Fingerprints Secure Nothing,” *Hackaday*, 10-Nov-2015. .

[44] “Cryptosteel • Master of All Backups,” *Cryptosteel*. [Online]. Available: https://cryptosteel.com/. [Accessed: 08-May-2019].

[45] “Råd&Rön - Vart tog våra bankfack vägen?” [Online]. Available: https://www.radron.se/Artiklar/Vart-tog-vara-bankfack-vagen?fbclid=IwAR2mn94_pGxAjLN7M-OddjhNbgnxlb9dQHCaF7VMfgeHsvWjaDBB57Y905Y. [Accessed: 10-Apr-2019].

[46] “BankID Selling Banks.” [Online]. Available: https://www.bankid.com/en/kontakt/foeretag/saeljare. [Accessed: 08-May-2019].

[47] “Teknisk information.” [Online]. Available: https://www.bankid.com/bankid-i-dina-tjanster/rp-info. [Accessed: 08-May-2019].

[48] “Historia.” [Online]. Available: https://www.bankid.com/om-oss/historia. [Accessed: 09-May-2019].

[49] K. Waddell, “When Fingerprints Are as Easy to Steal as Passwords,” *The Atlantic*, 24-Mar-2017. [Online]. Available: https://www.theatlantic.com/technology/archive/2017/03/new-biometrics/520695/. [Accessed: 09-May-2019].

[50] “Re-Authentication - Facebook Login - Documentation,” *Facebook for Developers*. [Online]. Available: https://developers.facebook.com/docs/facebook-login/reauthentication/. [Accessed: 21-May-2019].

TRITA-EECS-EX-2019:565