Secure Hardware Cryptocurrency Wallet within Common Criteria Framework

A thesis submitted to the
Graduate School of Natural and Applied Sciences

by

Yasir Emre BULUT

in partial fulfillment for the
degree of Master of Science

in

Cybersecurity Engineering

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Cybersecurity Engineering.

APPROVED BY:

Prof. Ensar Gül .....
(Thesis Advisor)

Dr. İsa Sertkaya .....
(Thesis Co-advisor)

Asst. Prof. Ali Rıza Ekti .....

Asst. Prof. Muhammed Ali Aydın .....

This is to confirm that this thesis complies with all the standards set by the Graduate School of Natural and Applied Sciences of Istanbul Sehir University:

DATE OF APPROVAL: 24.07.2019

SEAL/SIGNATURE:

Declaration of Authorship

I, Yasir Emre BULUT, declare that this thesis titled, 'Secure Hardware Cryptocurrency Wallet within Common Criteria Framework' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 24.07.2019

Secure Hardware Cryptocurrency Wallet within Common Criteria Framework

Yasir Emre Bulut

Abstract

Bitcoin paper, published under a pseudonym Satoshi Nakamoto, opened a new era; cryptocurrencies, blockchain and distributed ledger technologies that are aiming distributed trust model. Even if there has been an ongoing extensive discussion both on the origin and the future about these developing technologies, number of products, studies and projects are increasing day by day. Among these the most crucial one is crypto wallets since the distributed trust and privacy preserving solutions are all relies on the underlying cryptographic primitives and the corresponding cryptographic keys. Almost all the cryptocurrencies require their users individually manage their own cryptographic keys or recommend use of cryptocurrency wallets. A cryptocurrency wallet or shortly, crypto wallet, has to generate and store one or more public-private keys and corresponding addresses. These keys authenticate corresponding transactions, hence any adversary who gains access to a wallet may seize all the assets secured with them. Therefore, cryptocurrency wallet solutions and products should be carefully analyzed and better to be certified if possible from the very beginning.

In this thesis, we mainly focus on to what extend and how a cryptocurrency wallet's security analysis should be pursued. In order to formally portray the analysis framework, we propose to follow the Common Criteria (CC) Evaluation framework. CC evaluation framework is a formal evaluation methodology. For this purpose, assumptions, risks, threats and security vulnerabilities of the wallets will be defined. Also, objectives showing how these threats will be countered inside the wallet and what kind of measures should be taken by the environment and users will be detailed. In order to understand the security requirements, blockchain technology and Bitcoin which is the leading cryptocurrency will be explained and cryptocurrency wallets will be classified. In addition, we are going to propose a secure hardware wallet design in terms of physical and logical requirements. Then, we will compare the proposed wallet with other hardware wallets on the market. We believe that this thesis may be basic resource for creating standardized CC documents such as Protection Profile(PP), Security Target(ST) etc. Furthermore, this study would be a brief source for cryptocurrency wallets' design, test and analysis phases.

Keywords: bitcoin, blockchain, common criteria, cryptocurrrency, wallet, security problem, objectives

Ortak Kriterler Çerçevesi İçinde Güvenli Donanım Kripto Para Cüzdanı

Yasir Emre Bulut

Öz

Satoshi Nakamoto takma adıyla yayınlanan Bitcoin makalesi dağıtık güven modelini amaçlayan kripto para birimi, blokzincir ve dağıtık kayıt teknolojilerinin ortaya çıktığı yeni bir çığır açmıştır. Bu gelişmekte olan teknolojiler hakkında hem kaynak hem de gelecek ile ilgili kapsamlı bir tartışma olsa bile, ürün sayısı, çalışmalar ve projeler gün geçtikçe artmaktadır. Bunlardan en önemlisi, kripto cüzdanlarıdır ki dağıtık güven ve mahremiyet koruma çözümleri, temel kriptografik ilkellere ve bunlara karşılık gelen kriptografik anahtarlara dayanır. Neredeyse tüm kripto para birimleri, kullanıcılarının bireysel olarak kendi kriptografik anahtarlarını yönetmelerini gerektirir ya da kripto para birimi cüzdanlarını kullanmalarını önerir. Bir kripto para cüzdanı veya kısaca kripto cüzdanı, bir veya daha fazla özel-açık anahtar ve ilgili adresleri oluşturmak ve depolamak zorundadır. Bu anahtarlar, kripto para işlemlerini onaylamak için kullanıldığından cüzdana erişen herhangi bir düşman, anahtarlarla güvence altına alınan tüm varlıkları ele geçirebilir. Bu nedenle, kripto para cüzdan çözümleri ve ürünleri dikkatli bir şekilde analiz edilmeli ve mümkünse en baştan sertifikalandırılmalıdır.

Bu tezde, esas olarak bir kripto para cüzdanının güvenlik analizinin ne kadar derin ve nasıl yapılması gerektiğine odaklanıyoruz. Analiz çerçevesini tasvir etmek için Ortak Kriterler (CC) Değerlendirme yöntemini takip etmeyi öneriyoruz. CC değerlendirme çerçevesi, resmi bir değerlendirme metodolojisidir. Bu amaçla, cüzdanların varsayımları, riskleri, tehditleri ve güvenlik açıkları tanımlanacaktır. Ayrıca, tehditlerin cüzdan içinde nasıl önleneceğini ve çevre ve kullanıcılar tarafından ne tür önlemler alınması gerektiğini gösteren hedefler ayrıntılı olarak açıklanacaktır. Güvenlik gereksinimlerini anlamak için, blokzincir teknolojisi ve önde gelen kripto para birimi olan Bitcoin açıklanacak ve kripto para cüzdanları sınıflandırılacaktır. Ayrıca, fiziksel ve mantıksal gereksinimler açısından güvenli bir donanım cüzdan tasarımı önereceğiz. Ardından, önerilen cüzdanı pazardaki diğer donanım cüzdanlarıyla karşılaştıracağız. Bu tezin, Koruma Profili, Güvenlik Hedefi gibi standartlaştırılmış CC dokümanları oluşturmak için temel bir kaynak olabileceğine inanıyoruz. Ayrıca, bu çalışma kripto para cüzdanlarının tasarım, test ve analiz aşamaları için başlıca bir kaynak olacaktır.

Anahtar Sözcükler: bitcoin, blokzincir, ortak kriterler, kripto para birimi, kripto cüzdan, güvenlik problemi, güvenlik hedefi

Acknowledgments

I would like to express my gratitude to my thesis advisor Prof. Ensar Gül and co-advisor Dr. İsa Sertkaya for their guidance and constant encouragement. I am very grateful for scientific advice, knowledge and many insightful discussions and suggestions. It has been an honor to work with them.

I would like to thank to the rest of my thesis committee: Asst. Prof. Ali Rıza Ekti and Asst. Prof. Muhammed Ali Aydın for their encouragement, insightful comments and questions.

I would also thank to TÜBİTAK BİLGEM for its opportunity to study this Masters Degree and contribution on my experience on IT Security.

A very special thanks goes to my dear wife and my family who motivated and encouraged me to finish this thesis.

Contents

List of Figures

List of Tables

Abbreviations

CC	**C**ommon **C**riteria
CTCPEC	**C**anadian **T**rusted **C**omputer **P**roduct **E**valuation **C**riteria
EAL	**E**valuation **A**ssurance **L**evel
FIDO	**F**ast **Id**entify **O**nline
FINRA	**F**inancial **I**ndustry **R**egulatory **A**uthority
FTP	**F**ile **T**ransfer **P**rotocol
ISO	**I**nternational **O**rganisation for **S**tandardization
IT	**I**nformation **T**echnology
ITSEC	**I**nformation **T**echnology **S**ecurity **E**valuation **C**riteria
MITM	**M**an **i**n **t**he **M**iddle
NFC	**N**ear **F**ield **C**ommunatication
NIST	**N**ational **I**nstitute **o**f **S**tandards **a**nd **T**echnology
OSP	**O**rganizational **S**ecurity **P**olicy
PC	**P**ersonal **C**omputer
PIN	**P**ersonal **I**dentification **N**umber
PP	**P**rotection **P**rofile
QR	**Q**uick **R**esponse
RAM	**R**andom **A**ccess **M**emory
SAR	**S**ecurity **A**ssurance **R**equirements
SFR	**S**ecurity **F**unctional **R**equirements
ST	**S**ecurity **T**arget
TCSEC	**T**rusted **C**omputer **S**ystem **E**valuation **C**riteria
TOE	**T**arget **o**f **E**valuation
UML	**U**nified **M**odelling **L**anguage
US	**U**nited **S**tates
USB	**U**niversal **S**erial **B**us

Chapter 1

Introduction

Recording transactions and exchanges have been an important issue since ancient times. A ledger is the solution to record transactions and exchanges so that anyone could easily access the records when needed. While providing availability by easy access, safety is another concern for ledgers against malicious people since it could have sensitive and personal information. There have been significant improvements in terms of safety. In finance, listing and adding entries describing each asset or transaction is called single entry bookkeeping. Besides, if there is separation between assets and liabilities, each entry has a match on the other side and the sum of each side is equal, this ledger is called double entry bookkeeping. Also, triple entry bookkeeping which is an extension of the double-entry bookkeeping has been used for centuries to enhance security. While single entry systems are open to forgery and understanding errors are hard to detect, double entry systems are also vulnerable to forgery if there is no verification or proof of recorded transactions [12]. Even though triple entry systems are not secure enough and require trusted and neutral third party, they are harder to dispute and more secure than the double entry systems [13].

Development of technology offers better solutions for traditional methods. Instead of using slow and risky systems, distributed ledger technologies offering low risk, efficient control and more reliability are becoming widespread. Computer technology innovation enables digital media to replace the paper. On the other hand, blockchain technology enables digital media to be shared across the network all around the world. Network participants contribute to the blockchain system and all changes are reflected throughout

the ledger network. The security is maintained cryptographically by miners who solve complex hash sequences. These contributions make distributed ledgers to be preferred for trustless payment networks without any intermediaries. It could be stated that distributed ledger technology was able to emerge after some technological developments and maturity of databases, networks, cryptography, e-commerce and others.

In order to use the blockchain ledger technology, users must have cryptocurrency wallets to keep their private keys safe and execute transactions. The private key is the owner's identity similar to the personal signature in the distributed network. While sending coins, the transaction record is signed by the private key to maintain validation of authenticity, integrity and non-repudiation. The receiver uses his own private key to decrypt the message which is encrypted by the sender with the receiver's public key [14]. Cryptocurrency wallets enable these operations and provide secure means for private keys.

1.1 Related Work

A large amount of research has been done on the blockchain technology since Nakamoto's publication [14]. In this section, we will list the previous studies related to our thesis. Firstly, the books and their contents will be explained. Then we will list master's and doctoral theses related to our subject. Lastly, we will mention about other works, white-papers and web site articles etc.

In a book titled "Blockchain Basics" published in 2017, the concepts of why the blockchain is needed, how it works, limitations and usage areas are covered [15]. Tiana Laurence, in her book published by John Wiley & Sons in 2017, covers different blockchain applications such as Bitcoin, Ethereum, Ripple, Factom and platforms such as Hyperledger, Azure, Bluemix and their industry impacts in finance, real estate, insurance, government and other industries [16]. Melanie Swan categorized the blockchain in her book as "Blockchain 1.0: Currency, Blockchain 2.0: Contracts, Blockchain 3.0: Justice Applications and Blockchain 4.0: Efficiency and Coordination Applications". Blockchain 1.0 means applications related to cash and digital payment. Blockchain 2.0 is contracts which is more extensive than cash transactions such as stocks, bonds, loans, smart contracts. Blockchain 3.0 is the applications beyond these and covers usage in government, culture,

health and science. Besides these, wallet services and personal crypto security are mentioned in different aspects [17]. "Mastering Blockchain" written by Imran Bashir could be the most detailed book covering blockchain technology, cryptography, coins, alternative coin and blockchain solutions [18]. Another book describing blockchain technology in details and guiding for developing new technology solutions is written by Bambara *et al.*and includes technology, business and governance use cases with the examples of different best practices [19]. Blockchain 101, one of the very few sources in Turkish, contains a detailed description of the basic concepts, wallet functions, blockchain application areas, platforms, application examples, challenges and risks as well as technical details of the cryptography and the blockchain technology [20]. There is not much books covering security of the blockchain technology. In his book, Vincenzo Morabito dedicated a section to the security of blockchain systems and explained architecture, layers and challenges [21]. Another book named "Bitcoin and Blockchain Security" written by Ghassan Karame and Elli Androulaki mentions about payment security and privacy, recent attacks, possible countermeasures, user privacy and security of the Bitcoin wallets [22]. There are many more books written about the blockchain, Bitcoin and alternative coins, programming and mining, but we focus on the underlying technology and security analysis of the cryptocurrency wallets.

There are many dissertations published but since this is a new technology and financial sector is more interested in the usability, the studies mostly focus on application development. Researchers are generally concentrated in coins, contracts and other financial application areas. There is a master's thesis written by Karl Wüst containing information on security of blockchain technologies. He has mentioned about some attacks and vulnerabilities in Ethereum and Stellar [23]. Also in his article called "Do you need a Blockchain" he inquires whether the use of blockchain is really necessary or not and in which cases which blockchain types should be used [24]. In his master's thesis and article with the same name under "Trustzone-backed Bitcoin Wallet" Miraje Gentilal proposes a secured Bitcoin wallet with TrustZone which is a technology developed to increase security. This technology enables logical separation of secure and non-secure environment with an extension of processors and system architectures. Proposed Bitcoin wallet aimed to be more resilient against dictionary and side-channel attacks [2]. In [25], Bitcoin security is examined through the transaction and production methods. Security breaches are examined but the security is mostly discussed on hardware parts. Also,

this work includes suggestions about safe usage methods of wallets. Similarly, Bamert *et al.*also focused on hardware security and made a proposal of secure Bitcoin wallet named BlueWallet [26]. As a hardware token, this device, which communicates on Bluetooth to sign and authorize transactions, can be used with computers and smartphones and it is also expected to be used with cash registers.

There are lots of articles discussing blockchain applications and coins. Due to the increasing number of coin losses with attacks on wallets and the emergence of newly produced wallets claiming security, researchers began to work more on the safety of wallets. In an article about pervasiveness of blockchain, [27] mentions that data privacy problem has not yet been solved and this problem depends on the private key. Another article named "A Survey on Security and Privacy Issues of Bitcoin" discusses the challenges, risks and security considerations thoroughly. In a section discussing client side security threats, it is stated that wallet thefts are due to the use of system hacking, incorrect usage of wallet and buggy software installation. Management and secure storage of user keys are main points of relying on public cryptography on Bitcoin. There is also information about wallet types. Popular wallets are specified in a table in the aspects of type, interface, independence, underlying platform, privacy and security [28]. Low level communication security of the Bitcoin wallets are discussed by Gkaniatsou *et al.*[29]. They claim that their work is the first work stressing security issues of Bitcoin transactions in low level communication. Although they work on Ledger hardware wallet, it is stated that their security proposals could be adopted to all similar wallets easily. To analyze the communication protocol, they reverse-engineered the wallet implementation and found out that setup protocol could be accessed by attackers by forcing re-initialization, the plain-text PIN could be eavesdropped and all characters of the security card which is used for second factor authentication could be learned. They propose a lightweight fix by offering authentication protocol and sensitive data encryption. In a very recent article [30], hardware wallets are defined in a formal framework. Instead of manual inspection of wallet implementations, they aim to provide formal model of hardware wallets for verification by conceptualizing them as a system with different modules.

The studies about the physical attacks against hardware wallets are also published. Volotikin exposed the private key of second factor verification mechanism, proving that Ledger's flash memory is accessible [31]. In a presentation, Datko *et al.*explained how

they could extract the private keys with fault injection techniques and side channel analysis methods [32]. They used an open source tool to show the timing attack vulnerability on the processor used in Trezor and KeepKey hardware wallets. Before these works, known first physical attack to the Trezor hardware wallet was performed by Jochen Hoenicke [33]. He achieved to recover the private key by simple power analysis which requires less expensive tools and short time. His achievement provided developer to release an update including required patches.

There are also reports prepared by institutions and government agencies. A report prepared by National Institute of Standards and Technology (NIST) provides high level technical overview, detailed and useful information about the blockchain technology. It includes not only advantages but also disadvantages, limitations and misconceptions. Another point addressed in the report is that the blockchain technology has not been immune yet and there is still cyber security risks. Wallets are mentioned in the discussion of private key storage. It stated that there is a tendency to over-hype and overuse of the blockchain technology as in most developing technologies [34]. Section 3 of the report prepared by FINRA, a non-profit organization regulating broker-dealer operations, provides information on what to consider when creating a distributed ledger [35]. Briefly it declares that security should not only be thought as the only direct attacks on the system but also environmental safety, policies, assumptions and suggestions for usage should also be considered. Another report describing challenges for blockchain implementations is prepared by the Secure Technology Alliance. Security considerations section is the core part related to our thesis [36]. In "Blockchain and Cyber Security" report, Deloitte's experts discussed the security and maturity of blockchain in terms of the confidentiality, integrity and availability. They also addressed the authentication, authorization and non-repudiation specifications for new designs [37].

1.2 Contribution

Since blockchain technology is new and not yet immune as stated in [34], it may have several vulnerabilities. While there is an over-hype and race about cryptocurrency products, it is inevitable that security evaluations and considerations could be ignored or stay behind. Research and developments are mostly focused on analysis and formal abstractions about Bitcoin transactions and blockchain protocols as in [38], [39], [40], [41]. On

the other side, although there are individual efforts to analyze security of hardware and software wallets, currently no formal verification or any standardization of commercial cryptocurrency wallets has been published.

The main contribution of this thesis is to provide useful information for developers and users by examining the security weaknesses of wallets and fill the gaps of the formal evaluation methodology. Presented information would be collected as a resource for the PP documents used in CC, which is the most known international standard for evaluating the security of information technology products. Due to the fact that CC addresses product safety thoroughly and completely including environmental security, our proposal is expected to be enough for a wallet to provide security against known attacks.

In addition, the proposed hardware wallet in which we describe the design in our thesis will help producing secure wallets for developers. Comparing it with other hardware wallets we have shown that it is more reliable and secure.

1.3 Outline

This thesis consists of six chapters. In the first chapter, after the introduction section we included a literature review on the security of blockchain technology and cryptocurrency wallets. Then, we mentioned about the contribution of this thesis.

Chapter 2 gives information about blockchain and crypto wallets. Blockchain types, principles and structure are mentioned in here. After describing types of wallets and security considerations, we defined the assets of wallets to understand what needs to be protected. Lastly, working mechanism of a Bitcoin wallet is described.

Chapter 3 is focused on CC standard. After we briefly explain what is CC evaluation concept, evaluation levels, PP and ST documents, Functional and Assurance Requirements are described. We defined security problems of wallets in terms of threats, assumptions and organizational security policies. Also, this chapter includes security objectives which are protection rules against attackers and covers security functional requirements detailing security objectives defined in CC methodology.

Chapter 4 contains our hardware wallet design. We have combined all the aspects of the presented CC structure in this section and proposed a complete solution that provides overall security.

In **Chapter 5**, security analysis of known hardware wallets detailed with known attacks and prevention methods. Comparative analysis of hardware wallets is included to show the distinction between our secure wallet design and others.

Chapter 6 summarizes and concludes the thesis.

Chapter 2

Blockchain and Wallets

Blockchain technology was first introduced by Satoshi Nakamoto, a pseudonym, in his white paper named "Bitcoin: A Peer-to-Peer Electronic Cash System", proposed in 2008 and published on January 9, 2009. The technology and core idea behind the Bitcoin in Nakamoto's paper depends on the decentralized ledger and cryptographically validation of transactions instead of central authority [14]. His paper can be considered as a great innovative and revolutionary point in terms of related technology. In the meantime, many books and articles have been published about the blockchain. Most of these studies focused on financial systems and digital money. Besides this, blockchain could be used in many more areas such as smart contracts, copyright protection, digital identity etc. Decentralized type of ledger offers all participants to be able to view, monitor, log, approve and validate the records of transactions in a real time basis [42]. In this chapter we will give brief information about blockchain and wallets.

2.1 Types of Blockchain

The classification of blockchain technology can be done in different ways. In general, we see the classification according to approval requirement for participation to the network and read access to the blockchain ledger. Another method is approval requirement for participation in the reconciliation structure and write access to the ledger. In [34], categorization is done according to the permission model determining who is able to publish

a new block. They set the names as permissionless and permissioned networks. Permissionless blockchain networks let anyone to be able to publish new blocks, permissioned blockchain networks let only particular users publish new blocks. Also, permissioned ones can be exemplified as intranet in a corporate and permissionless ones as public internet which lets anyone to participate.

If all sides do not trust each other and there is a requirement for public verification, these type of permissioned blockchains are called as public permissioned blockchains. If there is a requirement for restriction of users, these type of blockchains are called as private permissioned blockchains [24].

In [15], same classification is done according to read and write restrictions. Another classification is done by Hileman *et al.*who separate blockchain as closed and open blockchains then separates open blockchain as public permissionless and public permissioned. Closed blockchains are divided into consortium and private permissioned blockchains [43]. It is obvious that same classification is defined with different names, public blockchain is called as open and private blockchain is called as closed.

2.1.1 Permissionless Blockchain

Permissionless blockchains let anyone to join and leave the network any time and there is no management or control for membership or unwanted entries. Some of the blockchain applications require this type of transparency so that any peer can read contents and verify blocks. Bitcoin, Ethereum and Zerocash are examples of permissionless blockchains [24]. Permissionless structure facilitates some functions of cryptocurrency wallets. Identification of wallet to the blockchain network is not required. Any wallet address could be used to send and receive coins.

Public permissionless blockchain, providing read and write access to everyone like Bitcoin, depends on less human contribution and more algorithmic security and data consistency [44].

2.1.2 Permissioned Blockchain

If there is requirement for permission to do any action on blockchain system, this type of distributed ledgers are called as permissioned blockchain. In [43], there is a brief history of the permissioned blockchain saying that the need of permissioned systems are required due to the corporate needs. Using public infrastructures run by anonymous users and unregulated structures caused some institutions to be uncomfortable. This need was met with the adaptation of permissioned blockchains to the enterprise requirements.

Permissioned blockchain networks may restrict and allow reading and submitting transactions. Consensus models could be used for publishing blocks without expense or maintenance requirements. There must be a level of trust for peers and if there is a misbehave, the authorization can be revoked [34].

Verification requirement in this structure could be achieved by verifying cryptocurrency wallet and corresponding owner. Handling every wallet and wallet address could be burdensome. Unlike independent structure of blockchain logic, this structure is reliant to the integrity of members or verification handlers.

There is an example of permissioned blockchain network called Open Blockchain originated by IBM. In this network there is a registration process for authorizing users and processing transactions. The system contains membership management responsible for identifying users, validating peers for transactions, non-validating peers for maintaining network and end users [22].

For the comparison of permissionless and permissioned blockchains, table 2.1 could be helpful to understand main differences in terms of read/write access, security, speed, identity and asset.

Table 2.1: Open versus Permissioned Blockchains [11]

	OPEN	PERMISSIONED
Read/Write	Open read/write access	Permission read and/or write access
Security	Compensate untrusted parties	Identified, pre-approved participants
Speed	Slower	Faster
Identity	Anonymous	Known identities
Asset	Native (e.g., Bitcoin, Ether)	Any asset

2.2 Principles of Blockchain

In [36], basic principles of blockchains are listed as the decentralization, trustlessness, consensus network, transaction transparency, transaction immutability and pseudonymity. These are listed as permissionless networks. Also there is another listing relatively short but added "survivable" item to these features [11]. In another paper, the most relevant properties of distributed ledgers are described and compared with centralized systems. Public verifiability, transparency, privacy, integrity, redundancy and trust anchor are the core elements of this paper [24].

Below is a combined list of blockchain principles.

Decentralization No central authority is required so that there is no single point of failure and vulnerability.

Trustlessness Trust is not needed in blockchain applications; due to the permissionless structure, everybody is allowed to enter and leave to the network.

Consensus Network For the validity of transaction, agreement over a decision is required, so there is a process and structure to do that.

Transaction transparency or Public verifiability In permissionless networks, all transactions are open to everyone.

Transaction immutability A transaction or block cannot be changed, damaged or deleted once it is added to the chain and validated.

Pseudonymous Transactions are done by anonymous nodes, nobody can see the identity of the peers.

Survivable Peers entering and leaving network does not harm the system or corrupt any data.

2.3 Structure of Blockchain

In this section, we will explain how the blockchain network works with the help of Bitcoin. In a typical blockchain network as seen in figure 2.1, there are participants who

are making transactions and validating these transactions. Also, there is a shared and distributed ledger to keep records of every transaction. User A sends coins to user B with his wallet that prepares transaction including sender, receiver and amount. This transaction is signed by user A to provide sender verifiability and non-repudiation. Every transaction prepared by wallets is sent to Bitcoin network. In this peer-to-peer network, blocks are established with the transactions to enable verification mechanism. A specific Bitcoin proof of work mechanism is used to verify each block. Once a verification is achieved, the block is broadcasted to all nodes. Receiver nodes verify each transaction in the block and accept the block as a new one. Each new block is added to the end of previous records and hash calculation is done to keep integrity. Recording the transaction to the ledger means that the receiver (which is user B in our example) received the coins.

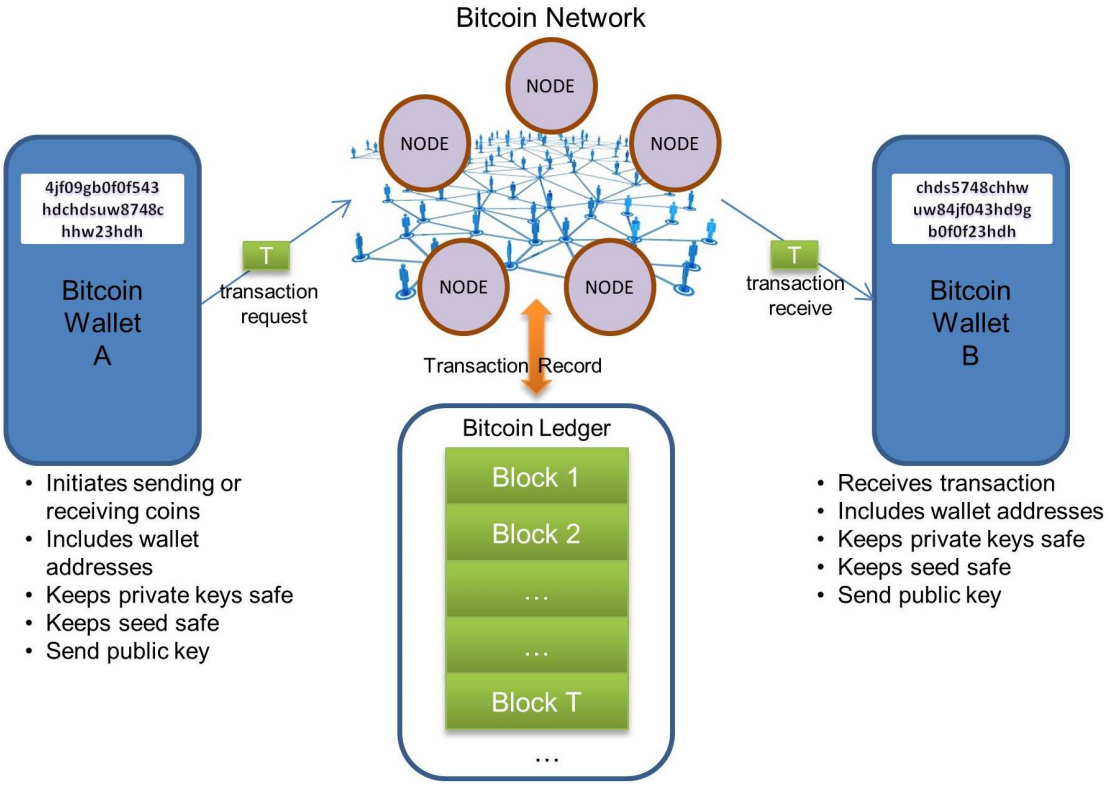


FIGURE 2.1: Structure of Bitcoin Blockchain Network

2.4 Cryptocurrency Wallets

Cryptocurrency wallets are required for the storage of addresses and public-private key pairs used for receiving and sending money. Wallets provide monitoring balances by

keeping track of transactions in the chain of blocks [45]. While some wallets are built for a single currency, most of the wallets can deal with several cryptocurrencies.

Simply, a cryptocurrency wallet is a software or combination of software and hardware using public-private key pairs. The design of the wallets may vary and users can choose suitable one according to their needs.

Specification of the wallets can be listed as following:

- Wallets store one or more public-private key pairs.
- Wallets store one or more addresses generated from public keys.
- Coins are not stored in wallets. The balance could be obtained from transactions.
- There are recovery methods against possibility of corruption or loss of wallets.
- Wallets could be working online or offline.

2.4.1 Types of Wallets

Hot and Cold Wallets: Primarily, we can divide wallets as hot and cold wallets. Hot wallets work always or mostly online and transactions can be executed at any time. Online (Cloud) wallets, desktop and mobile wallets could be specified in this type. On the other hand, cold wallets work offline and aim is to protect against attacks over the internet. Since network attacks are applicable mostly on hot wallets, they have wider attack surface. Cold wallets, generally refers to paper and hardware wallets are exposed to other types of attacks and it is questionable which type of wallet is safer. Stealing and losing are main security concerns of cold wallets [28].

Deterministic and Non-Deterministic Wallets: If the categorization is done according to the key generation methods, wallets are categorized into deterministic and non-deterministic [46]. Deterministic wallets are generating all keys from a seed which is called as single key or root key. There are mechanisms to generate each key pair from the seed. Seed is generated from a mnemonic sentence in case of failure or loss [47]. A mnemonic can bring all addresses and private keys while providing security. Since 512-bit keys are created from the seed, they can provide unpredictability in 2^{512} possibility. Non-deterministic wallet generates random and independent keys which requires backing

up all keys and storing [46]. Deterministic wallets are divided into three categories as deterministic, hierarchical deterministic and armory deterministic wallets according to their different security levels [48].

Another type of wallet classification is done according to satisfying user requirements and processing environment. Wallet types in this classification are called as **paper**, **mobile**, **desktop**, **online** and **hardware**.

Paper Wallets: Paper wallets are not used on their own. They are part of any other wallet application to keep addresses and keys safe physically. On paper wallets there are two QR codes; one is for encoding user's address to receive coins and other one is for encoding user's secret key to spend coins [22]. Transferring coins between hot wallets and paper wallet can be done when needed regardless of time. This process called sweeping and it could be done by scanning QR codes or entering private keys manually. One of the well-known examples of paper wallets is in figure 2.2.

FIGURE 2.2: MyEtherWallet Paper Wallet [1].

Mobile Wallets: While everything was going towards mobilization, it was inevitable for crypto wallets to be mobile. Providing accessibility, convenience and storing private keys locally; mobile wallets let owners to use them almost anywhere and anytime. Mobile wallets consist of applications running on mobile devices. They use another advantage of being hot wallet which is verifying transaction validity without downloading entire blockchain system [22]. Breadwallet, Coinomi, Mycelium, Toastwallet and Freewallet are some examples in this type.

Desktop Wallets: Running on PC or laptops, accessibility of desktop wallets is limited to the installed computer only. On the other hand, they offer a lot of features and services.

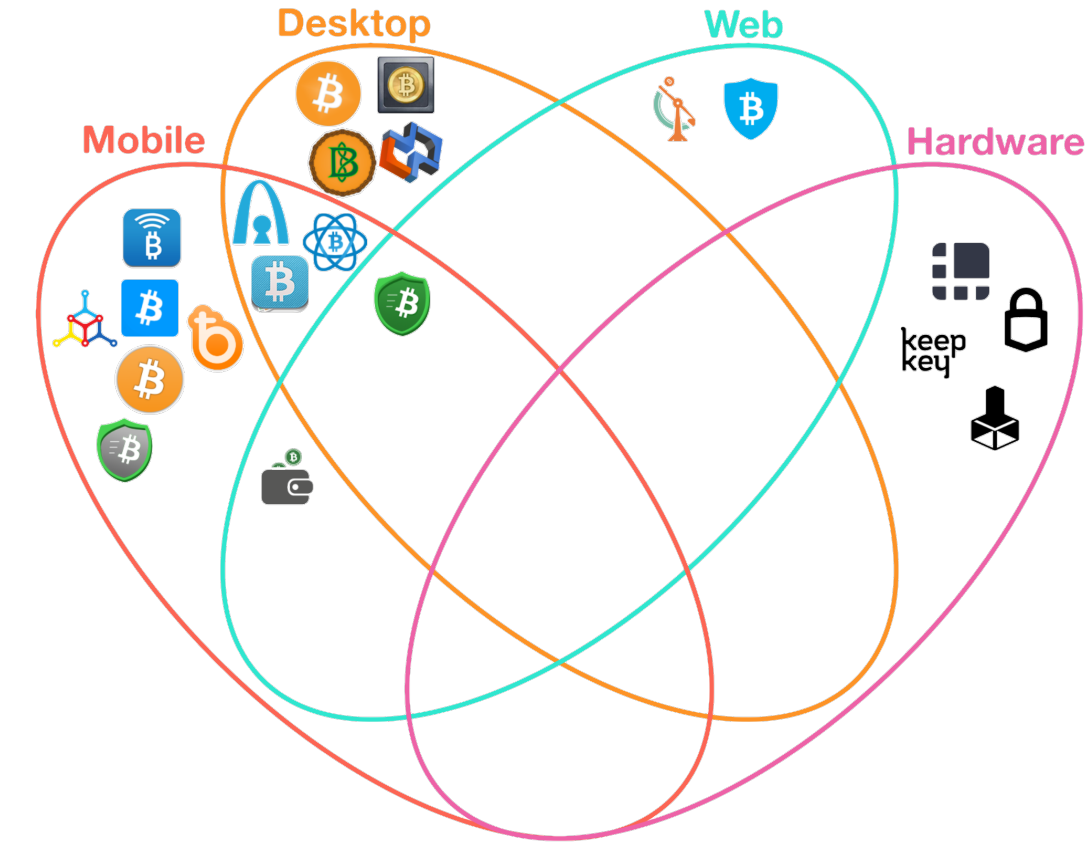


FIGURE 2.3: Crypto Wallet Brands Taxonomy.

Resource use and software capacity are higher than other wallet types. Aside from network attacks and virus threats, desktop wallets provide considerable security level [22]. Most used examples of this type are Multibit, Electrum, Armory, and Bitcoin/QT.

Online Wallets: Online wallets, also called as cloud wallets are web based wallets working on cloud systems. Keeping private keys in the cloud system makes them most remarkable target for attackers and prone to attacks. The most important features that distinguishes online wallets from others are that availability and accessibility conveniences [22]. In addition to the distributed structure of the blockchain, online wallets require trusting someone else; cloud providers. Coinbase, CoinKite, GreenAddress and SpectroCoin are some examples of online wallets.

Hardware Wallets: These are physical cryptocurrency wallets dedicated to securely store private keys and addresses. Offline storage of the sensitive keys makes these wallets more secure than other types. Online attacks are only expected during a transaction when a user connects hardware wallet to a computer . Besides this, this wallet type prone to specific hardware attacks and vulnerabilities. To prevent any attack, hardware devices

can be tested and certified according to some standards. The only condition for the security measures to be taken is that people should not lose their wallets [22]. Hardware failure is another issue about these wallets. This drawback, if there is no recovery policy or back up, can cause dramatic loss. For these reasons, corresponding precautions and security requirements are included in these devices. Keepkey, Ledger, Trezor and BitBox are the most known crypto wallets currently.

2.4.2 Security of Wallets

The main purpose of the wallets is to provide safe storage environment for private-public keys. The security functions of different wallets are based on the characteristics that are shaped according to the user needs and threats. Hardware wallets' security requirements are very different than other types of wallets. They are susceptible to hardware failures and theft besides hardware attacks. Software wallets are mostly prone to software failures and network attacks.

Security functionality of all type of wallets are mostly focused on keeping assets safe and providing secure authentication mechanisms. Proper methods for keeping assets during use or in storage are crucial. User authentication, password or PIN complexity and right implementation are also another essence parts of the security [49].

As in any other product, the security of the device alone is not enough and a holistic approach should be applied. For this reason, CC Methodology is applied to define security objectives for wallets which are called as Target of Evaluation(TOE) in CC and their operational environment.

About the authentication mechanisms, there are different ways enhancing security. Some wallets are using multi-signature (also called as multi-sig) authorization means requiring more than one key to authorize a transaction. Main purposes of multi-signature methods are establishing more security to prevent human error and creating democratic way to be used by one or more people. With this method, difficulty of the attacks are increased and probability of the coin loss is decreased [22]. We have seen examples that if any such measures were implemented before the attack, there would be no loss. The attack of Bitfinex which resulted in $65 million and Parity's loss of $30 million might be prevented [50].

2.4.3 Cryptocurrency Wallet Assets

In order to understand the security requirements, assets that need to be protected should be better explained. Contrary to the first thought that comes to mind, coins are not kept in cryptocurrency wallets, instead, blockchain ledgers keep records of coin transactions and this records prove the ownership of all coins. Assets of a cryptocurrency wallet can be briefly defined as protected objects, operations, security attributes and authorization data [4]. They are described in detail below.

Seed: A seed is used to generate key pairs. Since deterministic wallets use seed as a generator and chain of key pairs start from seed, this is the most important part of the assets. In case of a failure or loss, the recovery could be done by seed. On the other side, If this seed gets in the hands of the attackers or malicious people, the owner of the wallet loses all coins. In most of the wallets, seed is converted into mnemonic which is long sequenced word string created for users to remember easily. Human interaction with words or sentences are superior compared to the numerical representation of a seed [51].

Private Keys: Private keys are generated from seed in deterministic wallets by using a specific algorithm. Traditional wallets generate private keys randomly when needed. These keys are required to spend coins. During a transaction, wallet owner use a private key to sign it as a confirmation. This is also an important asset of a wallet. There could be a lot of private keys in a wallet and compromise of any key causes loss of corresponding coins.

Cryptographic Operations: Another asset to be protected in a wallet is the cryptographic operation used for exchange of coins. If there is a fault during hashing or asymmetric cryptography, an unexpected result could be obtained.

Security Functionality: Self-protection, secure initialization and non-bypassability requirements in a wallet are provided by security functionalities such as tamper resistance, tamper response mechanisms, obfuscation and software countermeasures. These are important specifications that need to work properly during the life-cycle of any wallet.

User Data: The authentication data of the owner is another asset in a wallet. Whether the password or PIN is in hash form or not, it should not be captured by attackers.

2.4.4 Cryptocurrency Wallet Working Mechanism

In this section, sending Bitcoin, transaction steps in a wallet and cryptocurrency network will be described. Receiving Bitcoin does not require any action from receiving party other than sharing public key.

Lets assume that Alice has converted her money to the Bitcoin and she has it in her wallet. She wants to send some Bitcoin to her friend Bob.

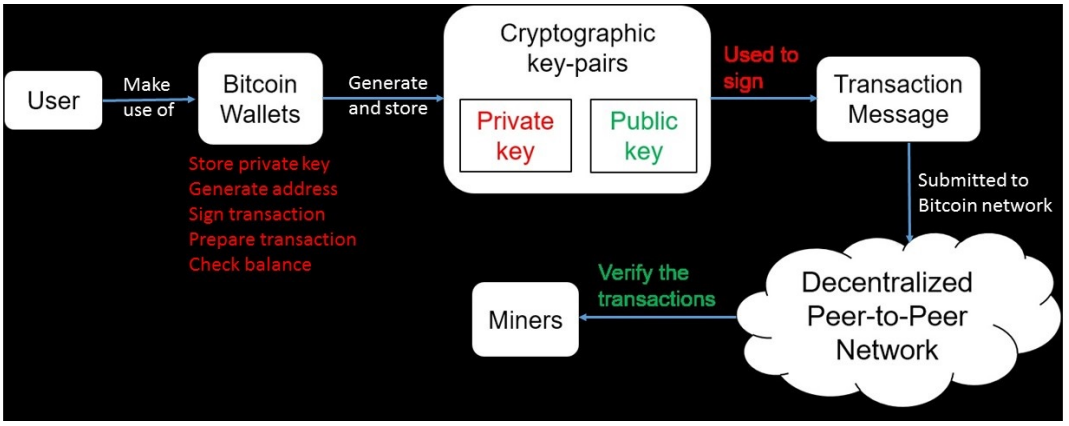


FIGURE 2.4: Transaction steps [2].

- First of all, Alice needs to log in to her cryptocurrency wallet. If it is a hardware wallet, after connecting it to a computer or mobile device she will log in on user interface. If she use desktop or mobile wallet, she just run the wallet application. In case of an online wallet, logging in is done on web interface. Authentication process must be secured by wallet and environment as stated in chapter 6.
- If there are more than one account in her wallet she chooses the account that she wants to use.
- She enters the amount of coin to send.
- Entering Bob's public address is also required. Public key enables Bob to receive Bitcoin and claim ownership. In this step, Alice is suggested to double check Bob's address against related attacks described in chapter 5.
- After filling every information, Alice transfers the coins to Bob. In this step, Alice's private key signs the transaction record which is prepared by wallet. Signing with private key provides Bob to verify the sender with the corresponding public key. Signing must be done in secure environment.

- When Alice submits the payment, the transaction is broadcasted to the Bitcoin network for verification process.
- This transaction record is included in a chain block together with other records. In Bitcoin network, transactions are bundled every ten minutes and miner nodes in the network verify the block with a special method.
- Miners calculate hash values according to a rule. When a miner achieves to calculate the required hash value, he earns Bitcoin prize and the transactions in the block are verified.

Chapter 3

Common Criteria

CC is developed in order to determine the security levels of information technology products and/or systems and to test them in independent laboratories. It is adopted by the International Organization for Standardization (ISO) as the International Criteria for Information Technology Security Evaluation Standard in 1999 (ISO 15408) [52]. CC combined and replaced European, US and Canadian criteria which are ITSEC, TCSEC and CTCPEC respectively. The importance and power of the CC comes from large involvement of experienced parties. The intention is to provide flexible and standardized security criteria, provide global recognition and acceptance. Once a product is evaluated and certified by a Certificate Authorizing Country, the certification is recognized by other countries. If the product will be sold in the international market, re-evaluation would not be required by each purchasing country [53].

Evaluation of IT products lets consumers to understand fulfillment of security features. At the time this thesis was written, there are 30 countries as Certificate Authorizing or Certificate Consuming participants and once a product is evaluated in one of these countries up to a certain level, the certification is accepted by all.

3.1 Common Criteria Definitions

Currently, the latest version of CC standard consists of three parts. CC Part 1 introduces general concepts and gives information about security evaluation. As an introduction and overview of the CC standard, this part includes description of other parts of the

standard, terms and abbreviations, general model, PP and ST specifications. Also, there is guidance for developers and evaluators about how to use the standard. Part 2 includes detailed definition of security functional requirements. Most common security requirements of IT products are listed in this part. These well defined requirements can be used by developers to establish trusted products. If requirements in this book do not meet the needs of developers, they are allowed to make own extended definitions by adhering to the general model. In Chapter 6 of Part 2, TOE, the subject of the evaluation, is defined as a set of software, firmware and/or hardware with the guidance documents [54]. A TOE might be any IT product of which assets or operations requiring security. Security Functional Requirements (SFRs) are pre-defined rules to control information and services in a TOE. CC Evaluation focuses on ensuring these rules are fully and correctly applied. In Part 3 of the CC Standard, there is Security Assurance Requirements(SARs). These are the measures defined in ST or PP to provide compliance with the claimed security functionality for each TOE.

3.1.1 Evaluation Assurance Levels

CC defines seven evaluation levels from EAL1 to EAL7 representing assurance packages which called as Evaluation Assurance Levels (EALs). These levels define how deep an evaluator will examine the TOE and related documents. The developer documentation must be detailed according to the evaluation level requirements. Depending on the assurance level, the deliverables include functional specifications, design specification, source code and guidance documents etc. [55]. As the level of evaluation increases, the examination detail also increases. Each level has a package of assurance requirements providing a strict level of detail and security. ST or PP writers choose a level and extend it with assurance components in case of an additional requirement. For example, a developer can choose EAL4 level with an additional component which provides higher vulnerability analysis detail [56].

EAL1 includes evaluation of functional and interface specifications of TOE and it provides security functions' analysis. Analysis is followed by independent testing of security behaviors of TOE.

EAL2 is the structurally testing level which provides analysis of TOE functions. High-level design of TOE subsystems are tested by using functional and interface specifications.

EAL3 requires more developer interaction and provides higher assurance for more serious threats. There is also requirement of configuration management and environmental control during development process.

EAL4 provides analysis of low-level design of TOE modules. Before testing, evaluators need to search up-to-date vulnerabilities independently. Development process controls are also supported by checking of life-cycle model, development tools identification and automated configuration management.

In an EAL5 evaluation, analysis of code implementation and semi-formal presentation of design is included. It is expected from the developer to provide modular and semi-formal design. Vulnerability analysis and calculation of attack potential require more robust production.

EAL6 evaluation focuses on modular and layered analysis of semi-formally described design and source code implementation. Vulnerability analysis is performed at high resistance level and rigorous development environment is required.

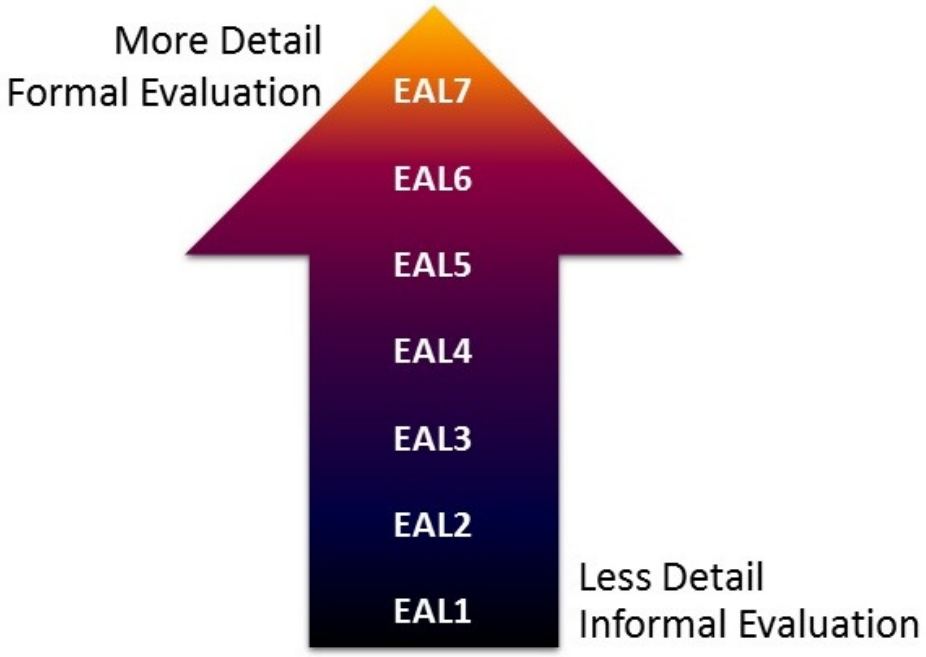


FIGURE 3.1: CC Evaluation Assurance Levels.

As the highest level of evaluation, EAL7 means formal presentation of functional specification and most detailed investigation of evidences. White box testing and independent

confirmation of developer test results must be done by evaluators. Due to the high risk situation and specialized security requirements, complexity of the product must be minimized.

Assurance through the evaluation is provided by analysis and checking of processes, procedures, correspondences between TOE parts, TOE design representation, guidance documents, functional tests, independent tests and potential vulnerabilities [56].

Other than these three part which constitute the CC standard, there is another ISO standard named Common Evaluation Methodology. This standard explains evaluation steps for evaluators and documentation context for developers.

In the CC evaluation process, stakeholders are evaluators, developers, sponsors and certificate authorities. Evaluators perform evaluation activities according to standards by examining documents, testing and analyzing TOE etc. [57]. First of all, the vendor prepares ST document describing security problems of the product, security objectives and security functional requirements. These requirements could be taken from PPs. Following the ST, vendor produces and prepares TOE and related documents for evaluation. If there is a sponsor, he is responsible for supporting the evaluation. Certificate authority is the main controller of the whole process, decision-maker and observer. Every report and test is controlled by the evaluation authority and based on the evaluation results authority issues certificates [58]. Evaluators' work is basically checking compatibility of the evaluation evidences which comprises TOE development, guidance, life-cycle definition and configuration management documentations, then testing the product functionally and finally doing vulnerability analysis [3].

3.1.2 Protection Profile

PP is prepared to address implementation-independent security requirements of a product type. It contains statement of security problem, functional and assurance requirements. The sections in this document must contain introduction, conformance claim, security problem definition, security objectives, extended component definition and security requirements.

In the introduction section, there is information about TOE type. Identification and abstract is given in the first part. Conformance claim section contains whether this PP claims conformance to any other PPs and/or packages.

Security problem definition section shows threats, organizational security policies(OSPs) and assumptions while security objectives section has solutions of security problems. Security objectives intent to counter security problems with TOE objectives and comply policies with environmental or TOE objectives. The trace between problems and objectives must be established completely and correctly. If there is any extended component they are included in extended component definition section. Extended means any components which are not included in CC standard.

Lastly there must be a section of security functional and assurance requirements which are translated versions of security objectives in specialized language of CC [59].

Security requirements can be determined by product or system purchasers in the PP documents and they can request vendors to claim conformance to a PP and satisfy defined requirements in the ST of the product [55].

3.1.3 Security Target

ST document identifies the security properties of a TOE. It is unique for each product. ST documents could be referenced to a related PP or developers can define security properties on their own.

The content of an ST document is similar to the PP. Instead of a product group, every specification is documented by considering a single TOE. Briefly, it includes security functional requirements and corresponding security objectives, security assurance requirements and evaluation scope [57].

In figure 3.2 the development process of Security Target document is described.

3.1.4 Security Functional Requirements

Security Functional Requirements specify detailed security rules for TOE functions to access and use of its resources. They are pre-defined definitions in the second part of

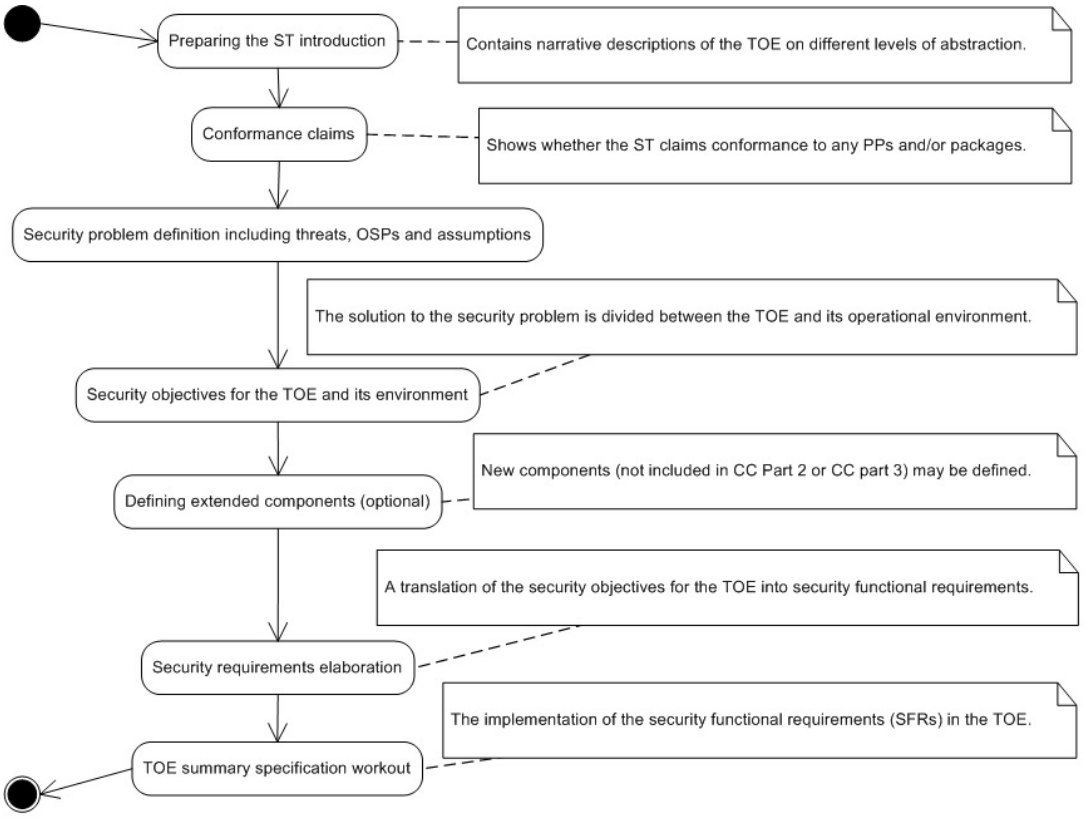


FIGURE 3.2: The CC-compliant IT security development process as a UML activity diagram [3].

CC standard. SFRs are grouped according to the specifications such as security audit, communication, cryptographic support, user data protection and security management. If a developer wants to define a specification about cryptographic operation, he can choose corresponding SFRs from the cryptographic support section.

CC allows developers to tailor SFRs through the use of permitted operations. The tailoring could be done by iteration, assignment, selection or refinement. Iteration allows developers to use the same SFR more than once for different operations. If an SFR has parameter setting field, assignment operation allows assigning the specification of parameters. Selection operation is used in multiple choice fields in the SFRs, developers can choose one or more items from the pre-defined list. If there is need of information addition or change for a detail in an SFR, refinement operation could be used.

3.1.5 Security Assurance Requirements

The measures that must be taken during the development and evaluation of a TOE defined by SARs. They provide assurance of compliance with the claimed security functionality. These components are used to determine what will be included in the evaluation. The assurance requirements for particular TOEs are documented in the ST generally as an EAL package and augmented components. SARs are defined in the third part of CC standard. Although almost all requirements are defined, additional requirements can be added by developers if required [56].

3.2 Security Problem Definition

Before going into problem definition, it is necessary to understand the assets within the TOE. In this way, the problem definition will be better understood because assets are the values that need to be protected in a product.

The assets of a cryptocurrency wallet were listed in chapter 2 as seed, private keys, cryptographic operations, security functionality and user data. Security problem is caused by the protection of these important assets. User data is used to identify the correct user of the system. Depending on the developers, PIN, password or multiple mechanisms could be used to authenticate the user. Protection of private keys used for signing in cryptocurrency transactions is also important. Since correct operation of wallets and security attributes provide proper and trusted work, they are accepted as assets. It is necessary to take precautions against such situations because it will cause suspicion in the security of the TOE.

Security Problem Definition section in a PP or ST includes the statement of security problems to be solved by the TOE and TOE environment. Threats, Assumptions and Organizational Security Policies are addressed in this section. Abbreviations at the beginning of the following definitions which are T, A and P denote Threats, Assumptions and Policies respectively. We have used CC standard and guidance documents to create these definitions.

3.2.1 Threats

In CC, threats are defined in terms of threat agent, adverse action/attack method and asset which is subject of the attack [4]. Threats are potential security vulnerabilities that have been created by considering the assets and interfaces. Situations that may disrupt the operation of the product and disable the security services are also considered as threat. Possible threats of cryptocurrency wallets defined according to CC terminology are as follows:

T.Compromise: The data in the protected area may be compromised by an attacker applying unauthorized actions [60]. This attack could be performed in different ways according to all wallet types.

T.UnauthorizedAccess: An attacker may perform adverse actions to bypass authentication mechanisms of seized, lost or stolen wallet. Attacker may gain root access of a wallet by bypassing PIN or fingerprint lock [49]. For biometric authentication, fake finger or several other ways could be used to fool fingerprint sensors. Cloud wallets are also vulnerable to these kind of attacks. This attack is different than the direct attacks such as brute force, dictionary etc. against authentication mechanisms.

T.ReverseEngineering: An attacker may obtain internal software design by reverse engineering to extract potential vulnerabilities. Vulnerabilities could be about hard-coded passwords, application specific information, security services and encryption keys [49].

T.FakeAddress: An attacker may alter the receiving and sending address to get coins into his own wallet. In such an attack, a trojan monitors the system always and it can change the sending address to the attackers address during the process of sending money [61]. For the receiving address, it can be replaced during the sharing of receiving address with a sender.

T.WeakAuthentication: An attacker may use related attacks to break the authentication mechanisms. These attacks could be brute force, dictionary, guessing user password, passphrase or PIN [49]. To enable this attack, hardware wallets need to be connected to a network or it must be seized by attackers, otherwise offline wallets are not target of this threat.

T.Eavesdropping: An attacker may monitor and listen the communication between application interface and wallets. This attack could be used to get authentication data or to get private keys if they are exchanged and used outside of secure area [49]. For hardware wallets, this attack is valid only when the wallet is used and connected to the online network.

T.DDoS: An attacker may use tools and/or botnet for online wallets to cause denial of service [62]. This attack degrades the quality of a connection or fully breaks it. The main intention of this attack is to make wallet services unable to be used. To execute this kind of attacks, attackers need to find a bug or weakness in the software implementation [63]. As stated in [64], this type of attacks can cause dramatic falls in coin prices. Even as in the previous examples some transactions sent for confirmation are blocked and some exchanges stopped the trade.

T.UnauthorizedUpdate: An attacker may try to update the TOE with malicious software and/or firmware in order to bypass security features and obtain sensitive data [65].

T.InformationLeakage: An attacker may perform non-invasive attacks which are also called as side channel attacks to obtain useful information. Useful data could leak during the cryptographic operations and attackers could extract private keys by analyzing leaked data. Power consumption of crypto processor, , timing information, electromagnetic emanation and input/output characteristics are the main sources of leakages [66] .

T.Hardware: An attacker may try to modify hardware parts of the wallets and get sensitive information or cause availability and authenticity compromise. Performing physical probing of the hardware parts and disclosing security functionality, authentication information and private keys are in this type of attack [67].

T.Malfunction: An Attacker may apply environmental stress to hardware wallets and cause malfunction in order to modify, deactivate or affect security services. Applying environmental stress means power, clock or electromagnetic glitches. The expectation of this attack is corruption in random numbers, security checks and control mechanisms and enable other attacks disclosing user data or manipulating software [67]. Before exploiting this attack, information about operational functionality or internal design must be obtained.

T.Reflashing: An attacker may install malicious firmware on hardware wallets to gain control over device [68]. Hardware based vulnerabilities could be used to reach and access firmware level of TOE.

T.Replacing: An attacker may replace a hardware wallet with a fake one to get access by obtaining authentication data. In this attack, attacker try to get owner's data by placing wireless transmitter or keylogger into the fake wallet [68]. Although replacing is not a direct attack to the wallet data, there could be mechanisms to overcome this attack in the wallets.

3.2.2 Assumptions

Assumptions are expectations from the TOE operational environment in terms of security aspects. Operational environment of a TOE could be users, hardware or software platforms, operating systems, other applications and physical locations. Meeting the expectations about environment strengthens secure functionality of the TOE. Threats are defined in normal operating conditions of the environment.

Since evaluation of environmental factors are beyond CC scope and CC is only suitable for IT systems assessment, assumptions cannot be tested during the evaluation process and directed to the operational environment.

A.SecurePlatform: All types of cryptocurrency wallets are assumed to be in a secure environment. It is expected that wallet platforms are working properly and securely. For each type of cryptocurrency wallets, the corresponding environmental components need to take necessary precautions. Untrusted applications come from unverified servers, malware, backdoor and rootkit installations are the main attacks that are expected to be prevented [49].

A.EducatedTrustedUsers: Authorized users are assumed to know recent cyber attacks and follow all procedures in user guidance not to expose any sensitive data. Also, users are expected to keep PIN, password and passphrases safe, check correctness of addresses and amount while sending coin, be careful about shoulder surfing. Recent cyber attacks include social engineering and phishing attacks [68].

A.SearchPoison: Search engines are assumed to take necessary precautions not to let poisoned search results which are redirecting users to the fake addresses. As stated in

the A.EducatedTrustedUsers assumption, users are assumed to be aware of this kind of fake address or phishing advertisement attacks and behave with caution. Since poisoning search result is not a direct attack to the wallet and there is no security objective for this situation, mitigation must be done by users. If users are redirected to the fake address, attackers can get their private information and drain wallets [69].

A.Update: Recovery and update assumed to be done in secure state and will not disrupt proper functionality of cryptocurrency wallets. During the update operation, any software other that newer version of wallet environment components should not be installed.

TABLE 3.1: Matching Threats, Assumptions and Organizational Security Policies(OSPs) with wallet types.

Threats/Assumptions/OSPs	Hardware	Mobile	Desktop	Cloud
T.Compromise	✓	✓	✓	✓
T.UnauthorizedAccess	✓	✓	✓	✓
T.ReverseEngineering	✓	✓	✓	
T.UnauthorizedUpdate	✓	✓	✓	✓
T.FakeAddress	✓	✓	✓	✓
T.WeakAuthentication	✓	✓	✓	✓
T.Eavesdropping	✓	✓	✓	✓
T.DDoS		✓	✓	✓
T.InformationLeakage	✓			
T.Hardware	✓			
T.Malfunction	✓			
T.Reflashing	✓			
T.Replacing	✓			
A.SecurePlatform		✓		
A.EducatedTrustedUsers	✓	✓	✓	✓
A.SearchPoison		✓	✓	✓
A.Update	✓	✓	✓	✓
P.StrongAuth	✓	✓	✓	✓
P.BackUp	✓			

3.2.3 Organizational Security Policies

Rules defined by an organization, authority or developer for a TOE to provide functionality and protect sensitive data are called as OSPs. These policies are expected to be enforced by TOE and its operational environment [57]. Specifications of mandatory security functions help mitigation of attacks and improve overall safety level.

P.StrongAuth: Robust and complex PINs, passwords and passphrases will be used to ensure a sufficient level of security. Authentication requirements defined by developer or customer should be stated in the operational user guidance document.

P.BackUp: In case of a possible failure, hardware wallets will be designed to have secure back up mechanism and provide recovery to maintain functionality and security.

3.3 Security Objectives

In order to provide a complete protection, security objectives are defined against security problems. These objectives split into two parts which are security objectives for TOE and security objectives for operational environment. In [57], security objectives are defined as "statement of an intent to counter identified threats and/or satisfy identified organization security policies and/or assumptions". Each security problem is addressed by at least one security objective. Since assumptions are defined for environment, they are only covered by security objectives for operational environment.

Security objectives for the TOE are assessed during the evaluation and further detailed in Security Functional Requirements(SFRs) section in ST documents. With the help of security objectives for operational environment which is providing correctness of environment, security objectives for the TOE counters threats.

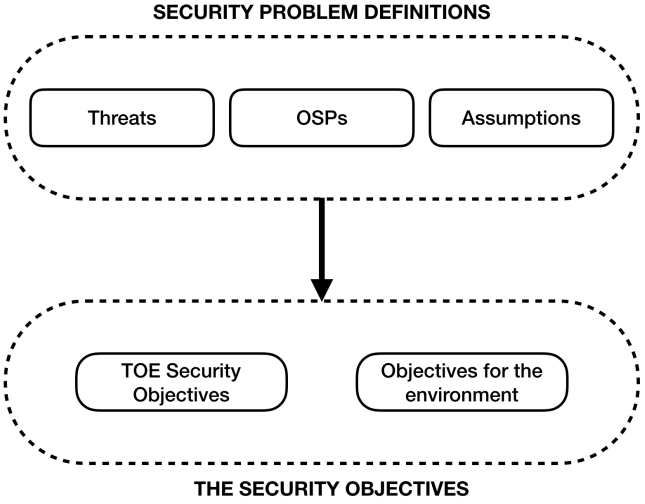


FIGURE 3.3: Role of the Security Objectives [4].

3.3.1 Security Objectives for TOE

OT.Access: A certain level of authorization mechanism and complexity is provided by security functionality of cryptocurrency wallets not to let attackers bypass these mechanisms and gain unauthorized access to the assets [67]. Two-factor or multi-factor authentication might be applied with a password, PIN or biometric data verification.

OT.ReverseEngineering: TOE security functionality provides protection against reverse engineering and does not let anyone obtain innate design to exploit possible vulnerabilities.

OT.FakeAddress: To prevent fake address attacks, cryptocurrency wallets operate protection mechanisms such as controlling and disabling system or clipboard monitoring.During the communication with the client, hardware wallets display the address on screen for verification. The user can compare sending or receiving address displayed on the computer screen and on the hardware wallet and then verify the transaction.

OT.Reflashing: Hardware wallets have protective and tamper resistant enclosure not allowing attackers install any firmware [67].

OT.Replacing: In case of replacing with a fake and malicious one, hardware wallets are designed in a way that is easily recognized by the owners. A unique message or figure could be displayed on the screen before authentication so that the user understands whether it belongs to him.

OT.WeakAuthentication: Wallets have security functionalities against authentication attacks and robustness against dictionary, brute force and guessing attacks. Long and complex passwords, passphrases and PINs, monitoring unsuccessful login attempts, CAPTCHA, enforcing retry time, delay and lock, are the main techniques of this objective [45].

OT.Eavesdropping: Communication between a hardware wallet and client software, an online wallet and user's computer, a mobile wallet and mobile platform or any other communication channel with sensitive information are subject to the eavesdropping attacks. Wallet security functionality provides obfuscated and encrypted communication against these type of attacks. Minimizing sensitive information outside the secure area is the most useful countermeasure in this regard [67].

OT.Storage: Data storage is protected in a secure way not to let attackers gain useful information in case of a compromise and leakage. Encryption of sensitive memory parts and address mixing provide adequate protection even if memory is dumped and data is captured.

OT.InformationLeakage: Hardware wallets are secured against information leakages arising from any kind of emanation. Sensitive data cannot be obtained directly or indirectly from the TOE [67].

OT.Hardware: Against hardware attacks, security mechanisms are implemented in hardware wallets to detect and block physical attempts. Tamper protections which are tamper evidence, tamper resistance and tamper response mechanisms prevent disclosing sensitive information and maintaining secure functionality [67]. Tamper evidence detects unauthorized intervention and ensures that the user takes the necessary action. Tamper resistance makes unauthorized attempts difficult or even impossible by applying suitable protection. Hardened cases and enclosures are examples for tamper resistance mechanism. Tamper response mechanisms are used to counter physical attacks via shielding or deleting sensitive data against disclosure. Tamper response includes detection and destroy the necessary parts.

OT.Malfunction: Hardware wallets are resistant against fault attacks caused by unexpected conditions. Fault attacks are implemented by physical sources like power, clock, electromagnetic glitches or laser and light beams. These invasive or non-invasive techniques aim to put the TOE in error state by fluctuations [70]. Necessary precautions (hardware sensors, secure software design etc.) should be taken since the inability to detect these attacks will affect the safety of the products [67].

OT.Audit: Audit trails are recorded to provide detection and prove evidence of hardware and software breaches. Keeping records play crucial role to understand failure, service discontinuity or unauthorized attempts.

OT.KeyCompromise: The design of a wallet is done in a way that keeps sensitive data in the secure area. Private key operations (signing a transaction) are done inside the wallet and there is no option to send or store these keys outside of the wallet.

OT.FailSecure: In case of a failure caused by any situation, wallets enter failure mode to maintain secure state. This is also called as fail-safe. When failure mode is securely

implemented, no attacker can bypass security mechanisms and obtain any information. This objective is very similar to tamper response mechanism in hardware attacks [60].

OT.Integrity: Wallets' security functionality has integrity check mechanism [60]. During a transaction, integrity check provides the correctness and uniformity of transaction information.

3.3.2 Security Objectives for Operational Environment

OE.DataImport: Data generation and transfer from outside to cryptocurrency wallets should be done with secure channels.

OE.Platform: Cryptocurrency wallets should run on reliable platforms. For this reason, the design and operation of the platforms should be organized in a secure manner against misuse, untrusted application, rootkit, malware or backdoor installation.

OE.Users: Wallet users are trusted and educated about how to handle well-known cyber security attacks and social engineering techniques [68].

OE.Components: Operational environment of a wallet is designed in a secure manner. Information leakage or faulty operation caused by underlying components are not expected [50].

OE.StrongAuth: Authentication mechanisms are used in a way that provides maximum security. PINs, passwords and passphrases are chosen at complex and unpredictable level. Repeated or sequenced numbers, simple letters or words must be avoided, random, robust and complicated but easy to remember passwords should be used [50].

OE.SafeSeed: In the event of a possible failure, the recovery seed or passphrases that restore the wallet must be kept secure. If the recovery seed or passphrase is captured by a malicious user, the wallet can be recovered and coins will be stolen easily [71].

OE.FakeAddress: Wallet users must be aware of forged search engine result attack and check the link and page before entering the authentication credentials.

OE.Update: Recovery and update of any environmental components are done in a way preserving secure state and normal working conditions. No environmental vulnerability are expected after the update resulting from this process.

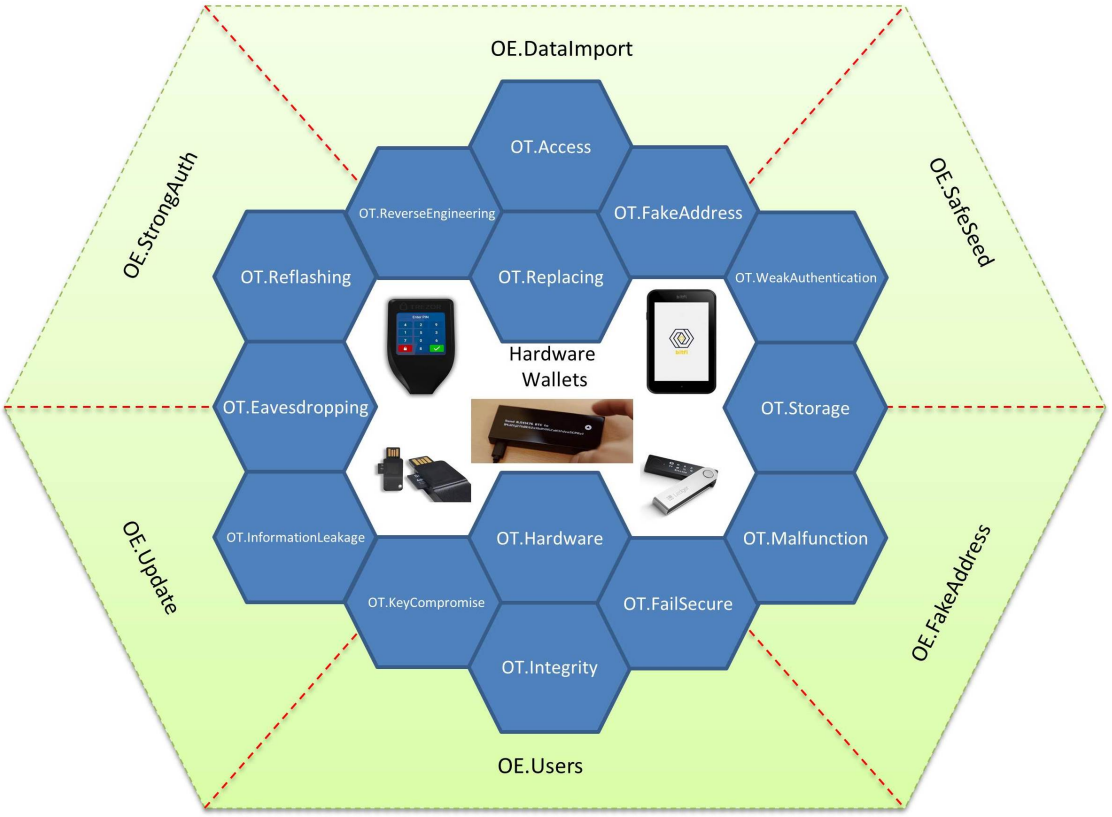


FIGURE 3.4: Proposed Protection Mechanisms of Hardware Wallets

The mapping between threat, assumption and OSPs are shown in Table 3.2. According to the CC, each threat is countered by at least one security objective for TOE while each assumption is uphold by at least one environmental security objective. Assumptions could only covered by environment. OSPs are enforced by at least one security objective.

TABLE 3.2: Matching Threats, Assumptions and Organizational Security Policies(OSPs) with Security Objectives.

Threats Assumptions OSPs	OT.Access	OT.ReverseEngineering	OT.FakeAddress	OT.Reflashing	OT.Replacing	OT.WeakAuthentication	OT.Eavesdropping	OT.Storage	OT.InformationLeakage	OT.Hardware	OT.Malfunction	OT.Audit	OT.KeyCompromise	OT.FailSecure	OT.Integrity	OE.DataImport	OE.Platform	OE.Users	OE.Components	OE.StrongAuth	OE.SafeSeed	OE.FakeAddress	OE.Update
T.Compromise	✓	✓				✓			✓		✓									✓			
T.UnauthorizedAccess	✓				✓	✓												✓			✓		
T.ReverseEngineering		✓				✓		✓									✓		✓				

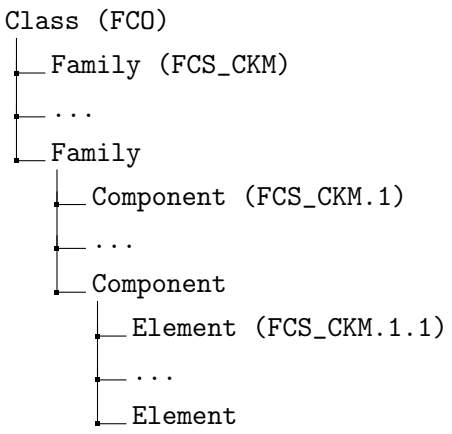
T.Reflashing				✓												✓		✓				
T.Replacing					✓												✓					
T.FakeAddress	✓		✓							✓					✓	✓	✓	✓				
T.WeakAuthentication	✓					✓				✓									✓			
T.Eavesdropping							✓			✓	✓			✓	✓		✓		✓			
T.DDoS							✓			✓	✓			✓			✓		✓			
T.InformationLeakage									✓									✓				
T.Hardware										✓		✓		✓	✓							
T.Malfunction											✓	✓		✓	✓							
T.UnauthorizedUpdate	✓			✓				✓							✓							
A.SecurePlatform																✓						
A.EducatedTrustedUsers																	✓					
A.SearchPoison																	✓				✓	
A.Update																						✓
P.StrongAuth	✓					✓													✓			
P.BackUp												✓										

3.4 Security Functional Requirements

Security Functional Requirements point out that how the security objectives are implemented in CC Methodology. SFRs in CC Part 2 are known and agreed criteria to create trusted products by defining the rules of access and use of TOE resources. Not all of the security requirements are covered in the CC and defined rules are not a definitive answer to all problems of TOE. Additional security requirements could be defined abide by the methodology [54].

Security Objectives will be covered in this section. Raw SFRs taken from [54] will be filled with necessary comments and additional information suitable for cryptocurrency wallets. There are many classes that cover different needs such as audit, communication, data protection and we will take the classes that match our needs and define the rest as extended.

Functional requirements are expressed in specific CC terminology; classes, families and components. Class section is used for unique identification and categorization with three characters. All functional requirements starts with the letter F indicating the functional Requirement. Other two letters are showing the class type. For the communication class the abbreviation of FCO is used. The same method is used as in family naming. Short name of a family is indicated in three characters. For the cryptographic support class and key management family, the naming is used as FCS_CKM. Functional class/family/component/element hierarchy is shown in the following tree:



3.4.1

Security Functional Requirements

SFRs are given as the same order in CC Part 2. Hierarchies and dependencies of SFRs are not added in this thesis, they are defined in [54]. If an SFR is hierarchical to another, it means that it provides more security functionality than the other. When an SFR is not sufficient or when it requires additional SFR for functionality, dependency arises. For example, cryptographic operation requirement needs generation or import of cryptographic keys.

3.4.1.1 Security Audit Class (FAU)

The abbreviation of FAU comes from combination of Functional Requirements and Audit Class. This class defines rules about auditing records.

FAU_ARP.1 Security alarms

According to this SFR, the TOE shall take required actions if a potential security vulnerability is detected. PP/ST author should define the actions to be taken according to the TOE type.

Considering crypto wallets for the SFR definition given in CC part 2 [54], these actions could be any warning messages, led flashing, sound or vibrating alarms to inform the owner to disable subject or functionality related to the potential vulnerability.

FAU_GEN.1 Audit data generation

This SFR defines auditable event requirements and information details in records for any incident audit records should be generated. Also, PP/ST author could define auditable events and detail level of records.

In an audit record, at least date, time and type of event, subject and event result must be included [54].

FAU_SAR.1 Audit review

FAU_SAR consists of two elements, one of these is used to determine by whom the audit records can be read. Second element states that the audit records must be suitable for user interpretation. Since we do not assume a user group and profile in a cryptocurrency wallet, service or maintenance user might read audit records. A wallet is dedicated to and owned by an individual user. Before any action, authentication is required but a limited channel could be provided for the service, maintenance or forensic reasons to the service or maintenance user roles.

FAU_SAR.2 Restricted audit review

Audit restriction is defined in this SFR component. Access to the audit records is prevented, except those allowed. If a developer wants to restrict read access of records for some users, this SFR must be added to the ST.

FAU_STG.1 Protected audit trail storage

Requirement of protection from deletion of audit records in audit trail is defined in the first element of this SFR. According to this SFR, wallets are designed to have security mechanism for prevention or detection of unauthorized modifications to the audit records. Protection of audit trail is very important for cloud wallets to understand if a security

breach happens.

3.4.1.2 Communication Class (FCO)

Identity of the originator and identity of the recipient of information transmitted in a data exchange, in other words proof of origin and proof of receipt mechanisms are assured in this class with two following families. Non-repudiation is provided by these components for both sender and receiver.

FCO_NRO.2 Enforced proof of origin

This SFR requires that evidence must be generated to maintain proof of origin for the user defined information. Also, requirement of association between attributes and these information is defined. Capability of evidence verification and restrictions must be defined by PP/ST author.

FCO_NRR.2 Enforced proof of receipt

This SFR consists of there elements. In the first element, proof of receipt will be generated for which information types is defined. Attributes of user defined information, capability of evidence verification and restriction requirements are defined in this section for proof of receipt.

Note: Proof of origin and proof of receipt are also part of the blockchain ledger. Since the transaction records are open to the public in the Bitcoin, anyone can read ledger and track the transactions. Blockchain structure requires transactions to be signed by sender in order to be checked by the receiver. Since signing process is handled inside the wallets, these SFRs are associated to the wallets and all of them provides these specifications.

3.4.1.3 Cryptographic Support Class (FCS)

In this class, security requirements for cryptographic functions are detailed. Key generation, key destruction and cryptographic operation details will be defined for cryptocurrency wallets.

FCS_CKM.1 Cryptographic key generation

Requirement for generation of blockchain related keys, algorithms, key sizes and corresponding standards are defined here.

For example, since Bitcoin uses Elliptic Curve Digital Signature Algorithm (ECDSA), key length of 32 bytes and curve which is secp256k1, algoritm, key size and the standard in which this algorithm defined must be listed here.

FCS_CKM.4 Cryptographic key destruction

This SFR is dependent on key generation SFR which is defined above because destruction is bounded to the production. To safely destroy sensitive keys, destruction methods could be defined according to a standard. Author can assign key destruction method and list related standards or define a new method in this requirement.

FCS_COP.1 Cryptographic operation

The assignment of which cryptographic operations are used by the wallets and which standards will be used in accordance with the operations are defined in this SFR. Cryptographic operation such as signing of transactions can be detailed with key size, algorithm and reference standard.

3.4.1.4 User Data Protection Class (FDP)

The requirement for authorization mechanism, complexity and specifications such as two factor, multi factor authentications, password and PIN usage are detailed in the following SFRs.

FDP_ACC.2 Complete access control

This SFR requires a user defined security functional policy about access control. In this policy, developers must define all possible operations of subjects on objects. TOE security functionality will cover all operations in this policy to protect user data during access control.

FDP_DAU.1 Basic Data Authentication

Verification of the validity or authenticity of information content during authentication is provided by basic data authentication SFR. One-way hash functions might satisfy this

requirement. List of objects or information types must be defined by PP/ST writer. According to the first element in this SFR, TOE security functionality will be capable of providing evidence for this user-defined information. In the second element, it is defined that which subjects will be able to verify the validity of evidence with the indicated information. This SFR is intended the protection of static data instead of transaction data.

FDP_IFC.1 Subset information flow control

Identification of information flow control policy is defined in this family. Beyond traditional mechanisms, non-interference policies, state-transitions, subjects, operations and the information under control of the policy are defined in this SFR. Level of information flow control policy could be defined either low level or high level. Since complete control of the information flow might not be required, only subset information flow control SFR is defined here. There is also complete information flow control SFR in CC standard but subset information flow control policy gives author more flexibility.

FDP_ITT.1 Basic internal transfer protection

If there is data exchange between physically-separated TOE parts, basic internal transfer protection SFR is used to define enforcement of access control and information flow control policies. The aim is to provide protection for disclosure, modification and loss of user data. If a physically-separated crypto wallet is designed by manufacturers, internal transfer protection can be provided by this SFR.

Since the protection of data flowing between TOE parts is guaranteed, internal transfer of sensitive information between parts of hardware wallets will be protected by this SFR. Also software wallets are protected by this SFR since they are installed on hardware components and similar protection mechanisms are applied for security of information flow.

FDP_ITT.3 Integrity monitoring

Access control and information flow control policies mentioned in the previous SFR is used here to monitor integrity of user data for the errors defined by PP/ST author. Also, any action could be specified to be taken upon the integrity error.

For example, if a user data integrity error is detected, wallet could be locked and no

transaction is allowed, even private keys could be erased and user is warned. Also, this process could be applied for other data protection SFRs.

FDP_RIP.1 Subset residual information protection

The protection of residual information is provided by this SFR. The definition of which objects are to be protected during allocation of the resource to or deallocation of the resource from must be specified.

FDP_SDI.2 Stored data integrity monitoring and action

In this SFR, TOE security functionality monitor stored user data and take necessary actions for integrity errors based on the user-defined attributes. Actions to be taken are also not defined in the CC standard and left to the PP/ST author.

FDP_UCT.1 Basic data exchange confidentiality

Confidentiality of data during transmit and receive is enforced in this SFR with the help of access control and information flow control policies.

FDP_UIT.1 Data exchange integrity

Besides confidentiality of data defined above, integrity of exchanged data must be enforced by access control and information flow control policies in a manner protected against one or more selection of modification, deletion, insertion and replay errors. TOE is expected to notice these kind of errors on receipt of user data.

3.4.1.5 Identification and Authentication Class (FIA)

Functional requirements for user identity verification are addressed in this class. Identification or authentication is used to ensure correctness of associated security attributes of users and it is important for the enforcement of security policies. FIA Class covers verifying and determining of user identity, associated roles or groups and authority on TOE. Other SFR classes are dependent on correct implementation of these requirements in order to be effective.

Authentication requirements are used in this part rather than identification, because wallets are generally personal and there is no multiple user distinction. In the blockchain,

identity of a user is defined by the address produced in the wallets. It is sufficient to authenticate the user in order to reach this address and perform transactions. Authentication requirements are defined below.

FIA_AFL.1 Authentication failure handling

In case of an authentication failure, TOE security functionality shall detect when an administrator defined number of unsuccessful attempts occurred. Also, the policy of required actions during the failure must be defined by PP/ST writer. The example of detection could be wrong PIN entry of certain times and the action could be adding wait time, blocking user for a period of time or a series of recovery steps. BitBox hardware wallet lets 15 failed attempts before unlocking device and after this level it erases all secret data [8].

FIA_SOS.1 Verification of secrets

TOE security functionality must provide verification of secrets mechanism meeting a certain quality metric which should be defined by developer. Quality metrics of user authentication passwords or PINs defined here.

FIA_UAU.2 User authentication before any action

Adding this SFR in a PP provides successful authentication requirement of user before any security functionality related action.

FIA_UAU.5 Multiple authentication mechanisms

Multiple authentication mechanisms are defined in this SFR to support user authentication. The use of this requirement will also become widespread as multiple authenticated wallets are being used. As a good example for this requirement Electrum wallet uses multisignature wallet mechanism and requires more than one key at the same time. Also two factor authentication which is using multi-signature mechanism increases security.

FIA_UAU.6 Re-authenticating

Re-authentication conditions could be defined in this SFR. If a developer want to re-authenticate user after each transaction, it must be stated in this requirement. These conditions may include idle time, number of transactions or any other conditions.

FIA_UAU.7 Protected authentication feedback

During the authentication, the feedback on the process might be sensitive. In such cases, the content of the data may be concealed in order not to release important information. Displaying star character instead of password, not displaying number of characters, feedback of general info instead of failed mechanism info could protect elements of this process.

3.4.1.6 Security Management Class (FMT)

We did not define any security requirement about security management functions. Since there is no control of users and attributes in cryptocurrency wallets, management system is not fully used. There is no user management or process management requiring security attributes. Owner or any user knowing the authentication information, PIN or password, can use the wallet and authenticate transactions. This class can be added and defined by ST authors according to further design details.

3.4.1.7 Privacy Class (FPR)

In this class privacy requirements can be defined on four ways; anonymity, pseudonymity, unlinkability and unobservability. Since Bitcoin is public and permissionless blockchain, there is no requirement of privacy. For users who want to be anonymous, there are some wallets such as Darkwallet providing anonymization by obfuscating transactions. Since wallets does not require such privacy to be more secure, these SFRs are not included in this work.

3.4.1.8 Protection of the TSF Class (FPT)

FPT class ensures the protection of integrity and management mechanisms of TOE Security Functionality (TSF) and corresponding data against tampering and bypass. TSF serves as the guard of the TOE. There is another protection mechanisms called FDP User Data Protection class but the difference between FPT class and FDP User Data Protection class is that one focuses on user data while other one focuses on TSF

data. Following SFRs define different aspects of protecting TOE security functionality and they required for hardware attacks defined in chapter 5.

FPT_FLS.1 Failure with preservation of secure state

This SFR ensures that in case of a failure, TOE will preserve secure state and will not lead to any compromise. List of failure types must be defined by PP/ST writer.

FPT_ITT.1 Basic internal TSF data transfer protection

According to this SFR, during the TSF data transmitting between separate parts of the TOE, protection must be maintained against disclosure and/or modification.

FPT_ITT.3 TSF data integrity monitoring

This SFR requires TSF to be able to detect and act upon one or more of modification, substitution, re-ordering, deletion of transmitting data between separate parts of the TOE. Also, the actions that shall be taken must be defined here upon detection of such integrity errors.

FPT_PHP.2 Notification of physical attack

TSF could be protected by physical aspects. First level of physical protection which is detection and notification of attacks is defined in here. TOE must detect physical tampering that might compromise the TSF and determine the occurrence level. After the detection, user must be notified about which device or element is tampered.

Hardware cryptocurrency wallets have to be protected against physical attacks. Notification of tampering will provide owner to be cautious and not to use vulnerable product. This level of protection ensures only the notification of attack. Users can be alerted by product-dependent features such as warning messages, sounds etc. Also tamper evident seals could be used on the physical junction parts.

FPT_PHP.3 Resistance to physical attack

Increased security provides resistance against physical tampering situations by automatic response. After the detection of defined attack scenarios, TOE will enforce security mechanisms and resist not to compromise any vulnerability. This requirement could cover not only invasive attacks but also semi-invasive and non-invasive attacks. Physical

barriers, hard enclosures, sensors could be used against direct physical attacks. Non-invasive attacks like power analysis could be prevented by other software and hardware methods.

FPT_RCV.1 Manual recovery

Another physical protection mechanism is when a failure or service discontinuity occurs, TOE enters a maintenance mode to maintain secure state. A service personnel or authorized intervention is required in order to check the system and return to the secure state.

FPT_RPL.1 Replay detection

During the communication, replay detection and other corresponding actions must be provided according to this SFR against attackers who are performing man in the middle attack. Clipboard copy protection may also be considered under this requirement.

FPT_STM.1 Reliable time stamps

This requirement defines the requirement of reliable time stamps. Since blockchain records use time stamping as proof, reliable functionality is important.

FPT_TST.1 TSF testing

Correct operation of a device must be controlled by self tests. Conditions and timing of self tests are defined in this requirement. Self tests could be implemented for TSF or TSF data at initial start-up, at user requests, during periodic controls in normal operation or at user defined conditions.

3.4.1.9 Resource Utilization Class (FRU)

This class has SFRs organizing fault tolerance, priority of services and resource allocation specifications.

FRU_FLT.2 Limited fault tolerance

Fault tolerance of wallets related to random number generation, flow of data, cryptographic operations and environmental faults like power supply are defined in this SFR. Errors detected by sensors also subjects of fault tolerance.

3.4.1.10 TOE Access Class (FTA)

This class contains the TOE access requirements for user session establishment. User session which is the interaction of user and system should be established according to identification and authentication attributes. Security must be considered at the beginning of the initial interaction. Following SFRs are defined for secure cryptocurrency wallets.

FTA_SSL.3 TSF-initiated termination

An interactive session must be terminated after a short interval of inactivity. If a user forgets to log out or turn off the wallet interface, this requirement will protect active session.

FTA_SSL.4 User-initiated termination

Besides automatic termination, TOE security functionality may allow user to terminate own session to maintain protection.

FTA_TAB.1 Default TOE access banners

This requirement can be used to generate a warning message if an advisory message or a confirmation sign is required. Wallets can display a user defined unique figure or phrase to let him confirm it is valid and correct. This requirement can protect wallet to be replaced with a fake one.

FTA_TAH.1 TOE access history

Display of unsuccessful session establishment attempts lets user understand when, where and how a malicious attacker tried to tamper the TOE. Also, successful session establishment information lets user see previous correct attempts, so that user can check whether someone else logged in or not. The access history is recommended not to be erased before reviewed by user.

3.4.1.11 Trusted Path/Channels Class (FTP)

While performing direct interaction with the TOE, users and other IT products need trusted path which is providing confidence. Trusted path protects from security breaches

resulting from untrusted applications. The conditions of secure communication channels are defined in the following SFRs.

FTP_ITC.1 Inter-TSF trusted channel

In this SFR, it is stated that protection of communication channel must be provided by TOE from modification and disclosure and the channel must be logically distinct from other channels. TOE or other IT products can initiate communication and PP author can define for which functions a trusted channel is required.

FTP_TRP.1 Trusted path

Similarly, TOE security functionality must provide secure communication against modification and disclosure. TOE shall permit and require users to initiate trusted communication and use it.

3.4.1.12 Extended Security Functional Requirements

Extended SFRs are defined additionally when the requirements in the standard are not enough to cover all functionality.

FPT_EMS.1 Emanation Security

Since there is no security requirement in the CC standard about electromagnetic emanation, we define the requirement in this section. During the cryptographic operations in hardware wallets, it is expected that malicious users cannot obtain useful information via physical emanations.

FCS_RNG.1 Random Number Generation

Random Number Generation is used for cryptographic operations. Quality metrics must be defined to obtain high quality random numbers. So that, functional requirement for this mechanism is defined in this family. PP/ST author could define which quality tests would be done, what will be the entropy level and what is expected from random number source. A recognized methodology e.g. AIS31 or ISO/IEC 18031 standard could be selected to comply with international standards.

Table 3.3 lists the relationship of Security Objectives with SFRs. Each SFR takes an important role to achieve the requirements of corresponding Security Objectives.

Table 3.3: Matching of Security Objectives and SFRs

Security Objectives	Related SFR ans SFR Description
OT.Access	FAU_GEN.1 Audit data generation
	FDP_ACC.2 Complete access control
	FIA_AFL.1 Authentication failure handling
	FIA_UAU.2 User authentication before any action
	FIA_UAU.5 Multiple authentication mechanisms
	FIA_UAU.6 Re-authenticating
	FIA_UAU.7 Protected authentication feedback
OT.ReverseEngineering	FAU_GEN.1 Audit data generation
	FDP_ACC.2 Complete access control
	FDP_ITT.3 Integrity monitoring
	FDP_SDI.2 Stored data integrity monitoring and action
	FIA_UAU.2 User authentication before any action
	FPT_ITT.3 TSF data integrity monitoring
	FPT_PHP.2 Notification of physical attack
	FPT_PHP.3 Resistance to physical attack
	FPT_RPL.1 Replay detection
	FPT_TST.1 TSF testing
OT.FakeAddress	FDP_IFC.1 Subset information flow control
	FDP_ITT.1 Basic internal transfer protection
	FDP_ITT.3 Integrity monitoring
	FDP_UCT.1 Basic data exchange confidentiality
	FDP_UIT.1 Data exchange integrity
	FPT_RPL.1 Replay detection
	FCO_NRO.2 Enforced proof of origin
	FCO_NRR.2 Enforced proof of receipt
OT.Reflashing	FDP_ITT.3 Integrity monitoring
	FDP_RIP.1 Subset residual information protection
	FDP_SDI.2 Stored data integrity monitoring and action

	FPT_ITT.3 TSF data integrity monitoring
	FPT_PHP.2 Notification of physical attack
	FPT_PHP.3 Resistance to physical attack
	FPT_TST.1 TSF testing
OT.Replacing	FDP_ACC.2 Complete access control
	FTA_TAB.1 Default TOE access banners
	FTA_TAH.1 TOE access history
OT.WeakAuthentication	FAU_GEN.1 Audit data generation
	FCS_CKM.1 Cryptographic key generation
	FCS_COP.1 Cryptographic operation
	FDP_DAU.1 Basic Data Authentication
	FDP_ACC.2 Complete access control
	FIA_AFL.1 Authentication failure handling
	FIA_SOS.1 Verification of secrets
	FIA_UAU.5 Multiple authentication mechanisms
OT.Eavesdropping	FCS_CKM.1 Cryptographic key generation
	FCS_COP.1 Cryptographic operation
	FDP_ITT.1 Basic internal transfer protection
	FDP_UCT.1 Basic data exchange confidentiality
	FPT_ITT.1 Basic internal TSF data transfer protection
	FTA_SSL.3 TSF-initiated termination
	FTA_SSL.4 User-initiated termination
	FTP_ITC.1 Inter-TSF trusted channel
	FTP_TRP.1 Trusted path
OT.Storage	FAU_STG.1 Protected audit trail storage
	FDP_ITT.3 Integrity monitoring
	FDP_RIP.1 Subset residual information protection
	FDP_SDI.2 Stored data integrity monitoring and action
	FPT_ITT.3 TSF data integrity monitoring
	FPT_PHP.2 Notification of physical attack
	FPT_PHP.3 Resistance to physical attack
OT.InformationLeakage	FDP_ITT.1 Basic internal transfer protection

	FPT_ITT.1 Basic internal TSF data transfer protection
	FPT_EMS.1 Emanation Security
OT.Hardware	FAU_ARP.1 Security alarms
	FPT_PHP.2 Notification of physical attack
	FPT_PHP.3 Resistance to physical attack
OT.Malfunction	FAU_ARP.1 Security alarms
	FAU_GEN.1 Audit data generation
	FDP_IFC.1 Subset information flow control
	FDP_SDI.2 Stored data integrity monitoring and action
	FIA_AFL.1 Authentication failure handling
	FPT_FLS.1 Failure with preservation of secure state
	FRU_FLT.2 Limited fault tolerance
	FPT_PHP.2 Notification of physical attack
	FPT_PHP.3 Resistance to physical attack
	FCS_RNG.1 Random Number Generation
OT.Audit	FAU_ARP.1 Security alarms
	FAU_GEN.1 Audit data generation
	FAU_SAR.1 Audit review
	FAU_SAR.2 Restricted audit review
	FAU_STG.1 Protected audit trail storage
OT.KeyCompromise	FCS_CKM.1 Cryptographic key generation
	FCS_CKM.4 Cryptographic key destruction
	FCS_COP.1 Cryptographic operation
	FDP_ACC.2 Complete access control
	FDP_IFC.1 Subset information flow control
	FIA_UAU.2 User authentication before any action
OT.FailSecure	FAU_ARP.1 Security alarms
	FAU_GEN.1 Audit data generation
	FDP_ITT.3 Integrity monitoring
	FDP_SDI.2 Stored data integrity monitoring and action
	FPT_FLS.1 Failure with preservation of secure state

	FRU_FLT.2 Limited fault tolerance
	FPT_ITT.3 TSF data integrity monitoring
	FPT_PHP.3 Resistance to physical attack
	FPT_RCV.1 Manual recovery
	FPT_TST.1 TSF testing
OT.Integrity	FAU_ARP.1 Security alarms
	FAU_GEN.1 Audit data generation
	FDP_IFC.1 Subset information flow control
	FDP_ITT.1 Basic internal transfer protection
	FDP_ITT.3 Integrity monitoring
	FDP_RIP.1 Subset residual information protection
	FDP_SDI.2 Stored data integrity monitoring and action
	FDP_UCT.1 Basic data exchange confidentiality
	FDP_UIT.1 Data exchange integrity
	FPT_ITT.1 Basic internal TSF data transfer protection
	FPT_ITT.3 TSF data integrity monitoring
	FPT_RPL.1 Replay detection
	FPT_STM.1 Reliable time stamps

The SFRs described in this chapter are matched in table 3.4 by selecting the appropriate ones for the hardware and software wallets. We grouped mobile, desktop and online wallets as software wallets because they have almost the same functional characteristics. Table 3.4 shows that all SFRs in the list are applicable to the hardware wallets. But it does not mean that one hardware wallet must implement all of them. Developers can choose some of them to increase functional specifications but they have to implement the ones that are related to the protection. In the next chapter, the need for protection will be explained by examples.

Table 3.4: Matching of SFRs and Wallet Types

SFR	**Hardware**	**Software**
FAU_ARP.1 Security alarms	✓	
FAU_GEN.1 Audit data generation	✓	✓

FAU_SAR.1 Audit review	✓	✓
FAU_SAR.2 Restricted audit review	✓	✓
FAU_STG.1 Protected audit trail storage	✓	✓
FCO_NRO.2 Enforced proof of origin	✓	✓
FCO_NRR.2 Enforced proof of receipt	✓	✓
FCS_CKM.1 Cryptographic key generation	✓	✓
FCS_CKM.4 Cryptographic key destruction	✓	✓
FCS_COP.1 Cryptographic operation	✓	✓
FDP_ACC.2 Complete access control	✓	✓
FDP_DAU.1 Basic Data Authentication	✓	✓
FDP_IFC.1 Subset information flow control	✓	✓
FDP_ITT.1 Basic internal transfer protection	✓	
FDP_ITT.3 Integrity monitoring	✓	
FDP_RIP.1 Subset residual information protection	✓	✓
FDP_SDI.2 Stored data integrity monitoring and action	✓	✓
FDP_UCT.1 Basic data exchange confidentiality	✓	✓
FDP_UIT.1 Data exchange integrity	✓	✓
FIA_AFL.1 Authentication failure handling	✓	✓
FIA_SOS.1 Verification of secrets	✓	✓
FIA_UAU.2 User authentication before any action	✓	✓
FIA_UAU.5 Multiple authentication mechanisms	✓	✓
FIA_UAU.6 Re-authenticating	✓	✓
FIA_UAU.7 Protected authentication feedback	✓	✓
FPT_FLS.1 Failure with preservation of secure state	✓	✓
FPT_ITT.1 Basic internal TSF data transfer protection	✓	
FPT_ITT.3 TSF data integrity monitoring	✓	
FPT_PHP.2 Notification of physical attack	✓	
FPT_PHP.3 Resistance to physical attack	✓	
FPT_RCV.1 Manual recovery	✓	
FPT_RPL.1 Replay detection	✓	✓
FPT_STM.1 Reliable time stamps	✓	✓
FPT_TST.1 TSF testing	✓	✓

FRU_FLT.2 Limited fault tolerance	✓	✓
FTA_SSL.3 TSF-initiated termination	✓	✓
FTA_SSL.4 User-initiated termination	✓	✓
FTA_TAB.1 Default TOE access banners	✓	
FTA_TAH.1 TOE access history	✓	✓
FTP_ITC.1 Inter-TSF trusted channel	✓	✓
FTP_TRP.1 Trusted path	✓	✓
FPT_EMS.1 Emanation Security	✓	
FCS_RNG.1 Random Number Generation	✓	✓

Chapter 4

Hardware Wallet Design within CC Framework

4.1 Secure Hardware Wallet Design

We have designed a hardware cryptocurrency wallet based on the protection mechanisms we have identified in table 5.1 and security objectives in chapter 3. In the following subsections, functionality of our wallet device is described as physical and logical scope.

Besides keeping keys in a protected area, this device is also designed to provide following services:

- Passwords Management,
- Personal Identifier and Authenticator,
- Contactless Payment Card,
- Data Encryptor working on USB or NFC connection,
- FIDO (Fast IDentity Online) Authenticator.

4.1.1 Physical Scope of Wallet Design

The wallet we have designed will have at least the following hardware features:

- Wallet will have overall physical protection. Hard metal enclosure will be used.
- Security sensors, detection switches as shown on figure 4.1 will be used.
- Screen, fingerprint reader, NFC and USB could be used for 2 factor authentication.
- Payment address will be shown to be 6 characters in a row so that the user can easily control it.
- Internal battery for sensors will provide continuation of security when main battery is run out.
- All sensitive key will be stored in secure area.

Secure Chip

From the functional requirements we defined, we can say that the heart of the physical scope is the secure chip. In addition to reducing security concerns, it also relieves designers' burden. Since most of the secure chips are evaluated and certified according to CC, it is convenient to choose one of those. Secure boot, secure memory, crypto co-processors and security sensors are some elements of those chips which makes them different. The design of secure chips primarily ensures resistance to attacks. If the resistance disappears, the response mechanisms against attacks are activated and the protected assets are zeroized to prevent from reaching the attacker. Although the chip is so safe, additional physical security measures need to be taken into account.

Communication Interfaces

Communication channels can be determined according to the needs of the users. We put Near Field Channel (NFC) and Universal Serial Bus (USB) interfaces into our design considering the future payment trends. Bluetooth and wireless communication are also other options that can be easily added. The NFC feature will be turned off when the device is not in use.

User Interface

We chose a large color touch screen in our design. The display can be redesigned according to power consumption and users' feedback.

Covers

The outer covers of the device are designed to be tamper resistant and tamper proof. In this way, the first measure will be taken against attackers. Covers will have sensitive tamper detection switches connected to the secure chip. If tampering is performed without the intervention of the service officer, it will provide evidence and detection. The design also has a led indicator and fingerprint reader on the front side cover.

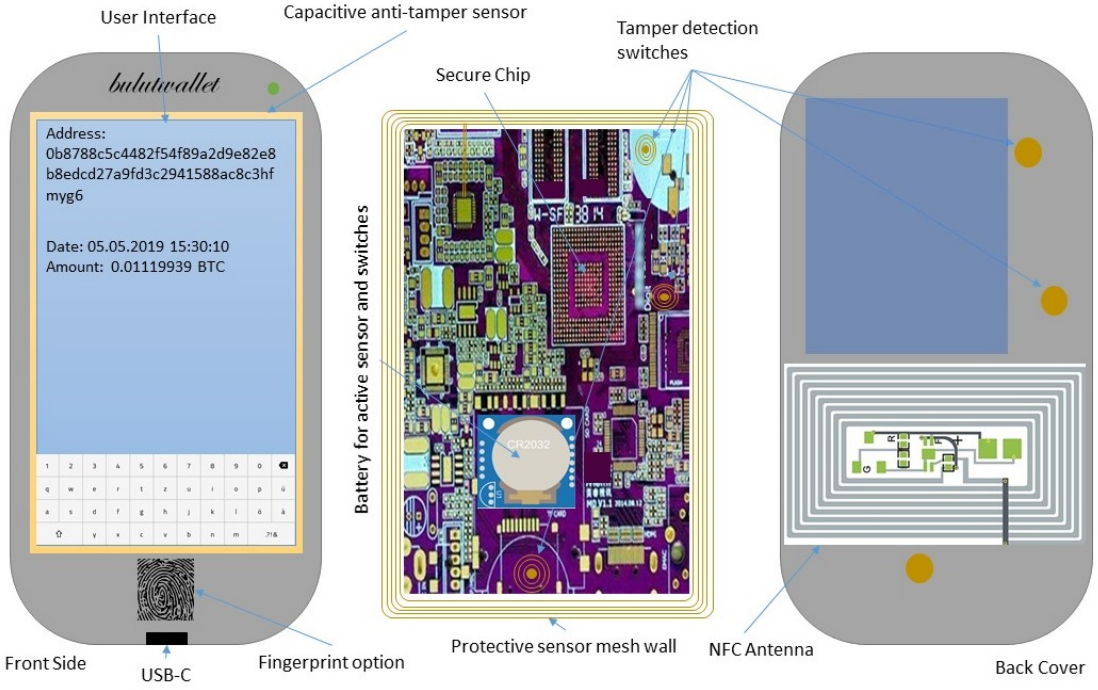


FIGURE 4.1: Physical Wallet Design

PCB Design

We have designed a safe area to protect the sensitive information. For this purpose, protective mesh sensor is recommended. It is an electrode layer consisting of conductive lines throughout the surface and connected to the secure chip. In case of any change of the signal on conductive lines, secure chip detects tamper. For an additional layer, epoxy based potting resin can be covered around the sensitive areas.

There is a small battery on the printed circuit board (PCB) for offline security. It will provide required power for mesh and tamper switch sensors when the main battery is run out.

4.1.2 Logical Scope of Wallet Design

The wallet we have designed will have at least the following logical features:

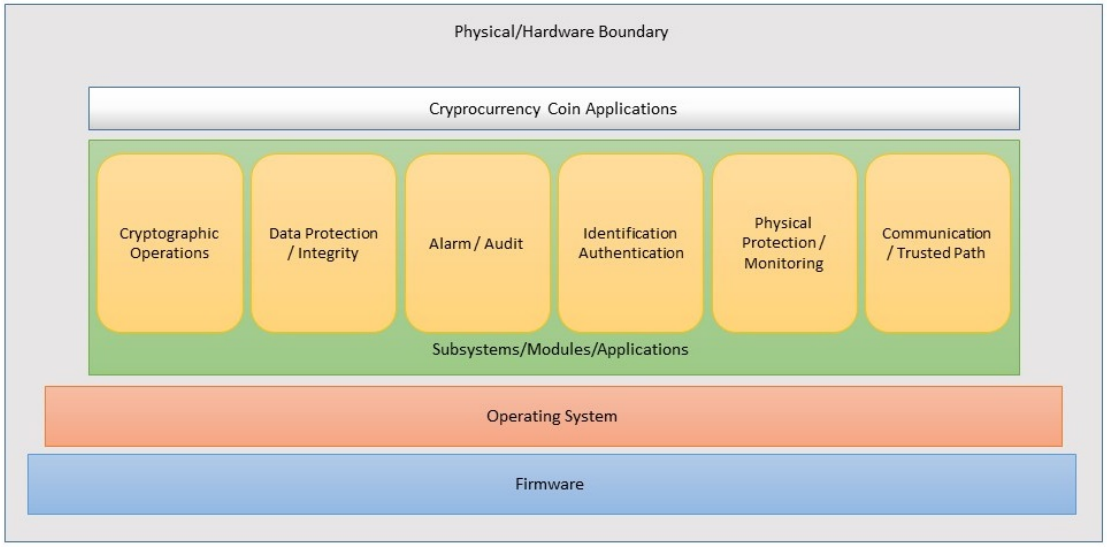


FIGURE 4.2: Software Wallet Design

- Audit data generation for every action, audit review restriction before authentication and storage protection.
- Generating cryptographic keys from internal true random number generator, destructing key and secure memory cleaning after expire. Applying constant time and adding random operations provide protection against side channel attacks.
- Requiring authentication for every action, asking at least 8 digits PIN and applying 30 seconds waiting time after 3 failed attempts, wiping the data after 5 wrong attempts. Re-authentication is also required for every coin transaction. On every start-up user will be notified with a pre-defined figure to show the validity of wallet and if there is a failed login attempt user will be warned.
- Data protection is applied on storage and during communication through authenticity, integrity and confidentiality. Obfuscation, encryption, power-on self-test, redundancy check, signature verification, memory management, replay detection, secure communication and address randomization techniques will be used.
- When a security breach or error is detected and if this situation is not in fault tolerance limits, owner will be notified and secure state will be preserved.

Cryptographic Operations

To achieve cryptographic operations required for blockchain transactions and authentications, every wallet device needs to implement certain algorithms. Since cryptographic

operations needs random keys, we generate unique random numbers in our secure chip (FCS_RNG.1). These random numbers are used to generate cryptographic keys, public-private key pairs, encryption keys etc. (FCS_CKM.1). These keys are used for signing/verifying transactions and encryption/decryption of user data (FCS_COP.1). There is also a key destruction feature for safe disposal of used keys (FCS_CKM.4).

Data Protection/Integrity

We have defined an access control security functional policy to determine scope of control. In our wallet, complete access control is implemented to ensure all operations and they are covered by the policy (FDP_ACC.2).

To protect transaction data, user is required to sign it. This functionality provides both basic data authentication and enforced proof of origin (FDP_ACC.2, FCO_NRO.2) Receiver can verify sender with the proof that is provided by sender (FCO_NRR.2).

During flow of information, security attributes need to be protected. For this reason information flow control functionality is included in the product (FDP_IFC.1).

To protect user data while it is transmitted between parts of the device there is another policy (FDP_ITT.1). Besides protection, integrity monitoring of user data is ensured. In case of an integrity error is detected, device may ask the data again or enter the failure mode to maintain secure state (FDP_ITT.3).

When a transaction is completed or a cryptographic key is destructed, there must be no residual information or it must not be accessible. There is a mechanism to ensure that resources are unavailable upon allocation or de-allocation (FDP_RIP.1).

The integrity of stored data is ensured by another functional requirement. Device monitors user data stored in the memory against identified integrity errors and takes necessary actions if a detection occurs (FDP_SDI.2).

To protect user data during a transfer between device and any application, security mechanisms of data exchange confidentiality and data exchange integrity are defined (FDP_UCT.1, FDP_UIT.1).

Alarm/Audit

Security alarm in case of a potential violation detection can be classified in this subsystem

(FAU_ARP.1). In our wallet, a warning message on screen will be displayed and no action will be available, only maintenance mode will be available to the technical service personnel.

This subsystem defines security audit functionality which includes generation and viewing of audit data (FAU_GEN.1 and FAU_SAR.1). Audit capability of device consists of recording system processes and user actions. Also in our design, audit review requires authentication mechanism (FAU_SAR.2). Audit trail storage will be big enough to keep every action in the wallet for one month and protection of stored audit data will be maintained (FAU_STG.1).

Identification/Authentication

Another functionality in this subsystem is authentication of user. Before any action, user must be authenticated (FIA_UAU.2).

In case of unsuccessful attempts reach a certain number, waiting time is applied. This will reduce the possibility of guessing attacks. Also the authentication will be blocked if 10 unsuccessful attempts are detected (FIA_AFL.1).

Strong PINs, at least 6 or 8 characters, will be asked by the system (FIA_SOS.1). There is an option to use multiple authentication mechanism so that user can choose multi-signature mechanism, fingerprint, verification by SMS or other mechanisms (FIA_UAU.5).

For every transaction, our design requires user authentication. Even if a user forgets to log off, the attackers cannot perform new transactions without passing authentication mechanism (FIA_UAU.6).

One of the authentication precaution is that the authentication feedback protection. While users entering PIN, it will not be displayed as plain-text on the wallet screen (FIA_UAU.7).

While device is turning on it will display a user defined message or figure to be ensure that device belongs to the owner (FTA_TAB.1).

Physical Protection/Monitoring

Physical protection mechanisms include not only protection of internal transfer but also

monitoring integrity of security functionality data (FPT_ITT.1, FPT_ITT.3). Compromise or altering of data flowing between physical parts of the device may cause malfunction and loss of assets. Device has a mechanism to detect replay of transmitting data (FPT_RPL.1). Also, in our wallet design, notification of physical attacks and resistance to the physical attacks are most important specifications to enhance reliability (FPT_PHP.2, FPT_PHP.3). Cold wallets, known as the most secure wallets, is offline so that the security of sensitive data is mostly relied upon physical protection mechanisms. Besides tampering, emanation is another issue that needs to be protected, so there are required physical and software measures in proposed wallet design (FPT_EMS.1).

In case of a hardware failure occurs or an attack to the hardware is detected and device cannot repair itself to continue normal operation, protection mechanism will preserve secure state and security sensors will continue to be active (FPT_FLS.1). It will ask for user intervention to return to the normal state (FPT_RCV.1).

This subsystem also protects functionality of time stamps. Since every process and transaction is recorded in the blocks of Bitcoin, a reliable time stamp is required (FPT_STM.1). Lastly, we can include self-tests that are a must for a system. Wallet will carry out self tests at start-up and periodically while working (FPT_TST.1).

Since availability is an important concern, our wallet is designed to continue working in case of acceptable errors. To achieve this, in the occurrence of identified failures, wallet will continue to normal operations. Failures are not related to security such as unavailability of communication channels or failure of additional applications etc. (FRU_FLT.2).

Communication/Trusted Path

Between user interface on desktop or mobile and device, there must be a secure communication channel against disclosure and modification of data. In our device, trusted paths are used for both initial authentication process and transaction establishment process. Secure communication will be established from the initial communication to the end (FTP_TRP.1).

We have defined a security policy to maintain trusted channel for updating the software and firmware of the device. Upon the mutual control of the certificates, both parties will allow to continue update. Only the device will initiate the update process with the

user's approval (FTP_ITC.1).

The secure channel between device and other parties could be terminated by device security functionality in case of an idle time is reached. If there is no user interaction for 1 minute the channel will be closed and re-establishment is required (FTA_SSL.3). Also, we allow user to terminate his own interactive session (FTA_SSL.4).

After the owner of the wallet turns on the device and confirms that it belongs to him, the first thing he will see is the previous unsuccessful attempts. Date, time and number of attempts will prove that malicious people made unauthorized attempts (FTA_TAH.1).

Chapter 5

Comparative Security Analysis of Cryptocurrency Wallets

In this section, we will analyze the real examples of security problems and categorize attacks. These real examples will reveal how important the security recommendations we offer. Then we will analysis cryptocurrency wallets comparatively according to the security objectives that we proposed.

5.1 Attacks and Prevention Methods in terms of Common Criteria

5.1.1 Malware Attacks

Bitcoin theft from a wallet is direct way of stealing with malware. As stated in a journal in 2011 [72], a trojan horse named Infostelar.Coinbit which attempts to steal Bitcoin wallets was discovered. This malware was searching Bitcoin wallet in windows operating systems. After locating the wallet, it uploads private keys using FTP to a remote server. It is reported that about 25000 Bitcoin were stolen with this method.

This was the first level of attack on low-security wallets. Although encrypting wallet data could increase security one step ahead and mitigate simpler attacks, it may not block

attackers since there are other ways of stealing. For holistic safety, security objectives defined for TOE and environment in chapter 3 must be applied.

Another way of stealing Bitcoin with the help of malware is to use clipboard hijacking. This kind of malware hijacks windows clipboard and replace the content. When a user copy a Bitcoin address, the receiving address will be different then the intended one. Symantec has discovered the first malware called Trojan.Coinbitclip on February 2, 2016 [73].

Related Threats: T.Compromise, T.UnauthorizedAccess, T.FakeAddress

Related Assumptions: A.SecurePlatform, A.EducatedTrustedUsers

Related OSPs: P.StrongAuth, P.BackUp

Related Security Objectives: OT.Access, OT.FakeAddress, OT.Audit, OE.DataImport, OE.Platform, OE.Users, OE.FakeAddress

Prevention Methods:

- Since the attack is targeting windows operating systems, TOE environment is expected to notify, resist and block such kind of malware attacks.
- Users are expected to be cautious not to share folders containing Bitcoin application [74].
- Instead of storing keys in local storage, password-protected wallets as stated in [74] could be used to mitigate such attacks. But in this case, attackers can use keyloggers to get users passwords and then encrypt files. Other precautions must be applied against these methods.

5.1.2 Unauthorized Access to the Hot Wallets

Hot wallets whet attackers appetite since they can be attacked at all times. Mt.Gox, an online largest currency exchange was hacked due to the lack of some serious security objectives such as version control software, testing policy and proper management [75]. Similarly, Bitcoinica, an exchange having always online hot wallets lost its wallets twice in three months [74]. In the first attack, a security breach at cloud service provider

Linode allowed attackers to steal about $210,000 from Bitcoinica. The second attack caused not only lose of $87,000 but also lose of client accounts and transaction history. Also it is commented that hackers captured webserver and reset the password.

Another hack due to the poor security measures happened to Coincheck, one of the biggest exchange in Japan. According to revealed details of the exchange's security, they were not using recommended multi-sig security and storing all currencies in a single hot wallet [76].

Related Threats: T.Compromise, T.UnauthorizedAccess, T.WeakAuthentication, T.Eavesdropping

Related Assumptions: A.SecurePlatform, A.EducatedTrustedUsers

Related OSPs: P.StrongAuth, P.BackUp

Related Security Objective: OT.Access, OT.WeakAuthentication, OT.Audit, OT.Eavesdropping, OE.DataImport, OE.Platform, OE.Components, OE.StrongAuth

Prevention Methods:

Both Mt.Gox and Bitcoinica attacks could be mitigated by two-factor or multi-factor authentication mechanisms. Also, storing large amounts in hot wallets is very high risk. Here are some of the recommendations for cryptocurrency users :

- Storing more Bitcoin outside of cold storage than one can afford is not recommended.
- Depositing to the cold storage is better way of saving if cold storage is protected correctly.
- From cold storage to the hot storage coin transfer should be done manually.
- Small amounts of withdrawals must not be ignored, it could be an attacker disguising a theft.
- Backing up the database to a secure place could protect from modification or deletion.

5.1.3 DDoS Attacks

The purpose of Distributed Denial of Service (DDoS) attacks are filling the bandwidth or resource of a targeted system with multiple users, not to allow legitimate parties to access the site or service. According to a report released by Imperva Incapsula, Bitcoin was one of the top-10 most targeted industries [77]. This attack may not be directly related to theft of coins but the effect of DDoS attack could be worldwide and may lead to a fall in prices as seen before [78]. In 2013, after a DDoS attack to the Mt.Gox, price of the Bitcoin went down from above $100 to the $55. At the end of 2017, Bitfinex, a cryptocurrency exchange went offline due to the DDoS attack. Severe losses were reported after this attack and some coin prices were decreased by as much as 90% [79]. Other attacks such as Bittrex and Bitcoin Gold have also been reported. It should be noted that only cloud wallets and exchange platforms can be attacked by this method but the effect could be universal.

Another version of DDoS attacks is mentioned by Gkaniatsou et al. in [29] saying that it can also be applied at command layer by tampering the transaction data and consequently blocking hardware dongle interpretation and transaction verification.

Related Threats: T.DDoS

Related Assumptions: A.SecurePlatform, A.EducatedTrustedUsers

Related OSPs: P.BackUp

Related Security Objective: OT.FailSecure, OT.Malfunction

Prevention Method

- In case a DDoS attack is detected, firstly the type and source must be identified. Then, related mitigation techniques must be applied.
- Applying custom web application firewall rules, using traffic monitoring tools could help to reduce the affect of attacks.
- Platform and network system must be established enough to maintain high level of traffic.

5.1.4 Phishing Attacks

In these attacks, the attacker sends an e-mail that imitates a person or an organization that requires the victim to click on the link, thus allowing the attacker to access critical data or information. If a victim clicks on the fake link, a malware could be downloaded to desktop or mobile platform. Hot wallets are more prone to this kind of attacks. In 2015, The Information Security Manager of Bitstamp clicked on the link in these messages and downloaded malicious software to the computer. As a result, the stock market was hacked and 19,000 BTCs were stolen at 5 million dollars on time . After this event, Bitstamp partnered with BitGo for multiple signature protection on crypto money transactions and transferred 98% of its digital assets to cold wallets [80].

Related Threats: T.Compromise, T.UnauthorizedAccess, T.ReverseEngineering, T.FakeAddress, T.WeakAuthentication, T.Eavesdropping, T.UnauthorizedUpdate

Related Assumptions: A.SecurePlatform, A.EducatedTrustedUsers, A.Update

Related OSPs: P.StrongAuth, P.BackUp

Related Security Objectives: OT.Access, OT.ReverseEngineering, OT.WeakAuthentication, OT.Eavesdropping, OT.FakeAddress, OT.FailSecure, OT.Audit, OE.DataImport, OE.Platform, OE.Users, OE.FakeAddress, OE.Components, OE.StrongAuth,

Prevention Method

- The best method of protection from these attacks is to educate the users to take adequate precautions.
- Also, the system would be designed with up to date intrusion prevention and detection systems.
- Control and restriction of admin and user rights must be defined.
- Using cold wallet and multi-signature mechanism could provide significant security.

5.1.5 Man In The Middle Attacks

Man-in-the-middle attack (MITM) is based on the principle of seeing and capturing the data between sender and receiver. If the attacker only watches the traffic and do not changes anything, this is called passive attack. If the transmitting data is altered and forwarded, this is called active attack. Fake address threat is defined according to this attack.

In terms of cryptocurrency wallets, MITM is applicable to mostly online (hot) wallets. The interception is more possible in these systems. But as mentioned in [29] by Gkaniatsou et al., through the sniffing communication between API and dongle, obtaining keys could be possible. They successfully obtained wallet seed in plaintext which is transmitted during setup process and able to regenerate master private key by using known derivation function. Setup process and reinitialization could be started by forcing wrong PIN verification attempts. Also they discovered that in Ledger Nano wallet, at each login attempt the PIN is transmitted as plaintext and if it is eavesdropped, it could be captured. By applying active MITM attack, Gkaniatsou et al. proved that tampering security properties of wallets, learning second factor authentication security card characters and altering the payment account and address could be possible. To be able to do these kind of attacks, adversaries have to gain authorization on the network and computer initially.

Hot wallets are also prone to these attacks since attackers can reach more interface to use network communication channels. If network security is not fully ensured, these attacks and Bitcoin loses will be inevitable.

Related Threats: T.FakeAddress, T.Eavesdropping

Related Assumptions: A.SecurePlatform, A.EducatedTrustedUsers, A.SearchPoison

Related Security Objective: OT.FakeAddress, OT.Eavesdropping, OE.DataImport, OE.Users, OE.Components, OE.FakeAddress

Prevention Method

- Preventing a MITM attack against replacing the sender/receiver address on a hardware wallet is relatively easy. As we have defined an SFR FTA_TAB before, TOE

could display the address on it and user can check the copied address with the displayed one.

- Other MITM attacks could be noticed by integrity and confidentiality protection SFRs.
- Communication channels must be encrypted and strong encryption must be used. Using virtual private networks might be an easy way of protection.
- Authentication mechanism must be applied before sending any information.
- Hot wallet platforms must have intrusion detection systems monitoring network and traffic continuously.

5.1.6 Hardware Attacks

Reflashing of hardware wallet memory, tampering of hardware enclosure, storage and malfunction attacks could be grouped under this title. Hardware related attacks will be discussed with the experienced examples in the following paragraphs.

Most known hardware wallets which are Trezor One, CoolWallet S, Ledger Nano X, Ledger Nano S, Ledger Blue, KeepKey and Bitbox have different design and security countermeasures. Trezor is the first wallet introduced in 2011 while CoolWallet S is produced in 2014.

As an example for direct hardware wallet attacks we can show that Ledger Nano S exploitation which is revealed about a year ago. 15 year old Saleem Rashid discovered that anyone can update the Nano S with the malicious software before generation of the seed. He took advantage of non-secure processor which is handling operations that do not require security. Despite there is a secure processor, it does not realize the malicious code. When the wallet is sent to and used by a user, the attacker can capture everything [81].

In 2015, a side channel attack is extracted Trezor hardware wallet's private key. In this attack, attacker used simple methods of side channel attacks [33]. Based on this attack Datko *et al.*developed another attack method and presented that they could achieve to obtain private key with timing attack vulnerability [32]. Their attack targetted a common processor which is used in Trezor and KeepKey.

Volotikin showed that most protected parts of hardware wallets could be exploited [31]. He explained that the ability to create and install wallet application for new cryptocurrencies cause vulnerability and an attacker can create malicious wallet application on hardware wallets. If there is not secure isolation, application can read data from other containers. He also discovered that this problem could be caused by resetting. Ledger Nano S does not wipe users' private data and flash is not cleared upon device reset. He suggests that all syscall parameters need to be checked and restricted and third party evaluation is a must for secure solution.

About hardware tampering, it is stated in [82] that Ledger Nano S does not have tamper evidence case, also debug port is directly accessible. This means that it is prone to any kind of attacks.

Another Bitcoin hardware wallet Bitfi was examined by two researchers and keys are extracted [83]. It is discovered that rooting the device does not clean memory and they can extract secret phrase and salt which are allowing private key generation.

One of the recent attacks is done on Trezor wallet to get PIN through the side channel attack. In this attack, the physical behaviour of the wallet analyzed and found out that power traces have leakage information. Attacks succeeded to find the correct PIN of a Trezor wallet with only 5.5 PIN attempts [84].

Related Threats: T.Compromise, T.UnauthorizedAccess, T.ReverseEngineering, T.Reflashing, T.Replacing, T.FakeAddress, T.WeakAuthentication, T.Eavesdropping, T.InformationLeakage, T.Hardware, T.Malfunction, T.UnauthorizedUpdate

Related Assumptions: A.EducatedTrustedUsers, A.Update

Related OSPs: P.StrongAuth, P.BackUp

Related Security Objective: OT.Access, OT.ReverseEngineering, OT.FakeAddress, OT.Reflashing, OT.Replacing, OT.WeakAuthentication, OT.Eavesdropping, OT.Storage, OT.InformationLeakage, OT.Hardware, OT.Malfunction, OT.Audit, OT.KeyCompromise, OT.FailSecure, OT.Integrity, OE.DataImport, OE.Users, OE.StrongAuth, OE.SafeSeed, OE.FakeAddress, OE.Update

Prevention Method

- To keep private keys safe, a user should make a copy and store in a protected place. This could be a paper wallet inside a safe or deposit box.
- Backup keys, passphrases or passwords must not be kept in online devices.
- Multi-signature mechanism keys must be placed separately. If they stay in the same place, they would be compromised together.
- Keeping large part of the coins in the cold storage and keeping daily operating part in a hot wallet helps mitigation of losses in case of an attack.
- Multi-factor and multi-signature mechanisms are highly recommended.
- Wallet owners must use strong passwords and unpredictable PINs.
- Wallet manufacturers must design hardware secure enough not to let attackers pass the security mechanisms. Tamper detection, tamper resistance and tamper response mechanisms must be used.
- Wallet mechanism should never send private keys outside or wallet never let any command to get sensitive information. Private keys and seed must be kept isolated.
- Latest version of the wallet software must be used.

TABLE 5.1: Experienced Attacks and Proposed Protection Mechanisms

Attack	**Victim**	**Loss**	**Reason**	**Protection Mechanisms**
Malware Attacks	All wallets	Unknown	Malware and Virus	Anti-malware, anti-virus software, Educated Users
Unauthorized Access	Mt.Gox	$473 Million	Disorganized Organization	Organized and Secure Management Structure, trusted platforms, hybrid hot/cold wallet system, multisig
	Bitcoinica	$210,000 and $87,000	Linode, cloud hosting platform compromised	
	NiceHash	$80 million	employee credentials to gain access	
	Coincheck	$530 million	single hot wallet, not using multisig contract security	
Implemantation Weakness	DAO	$50 million	loophole in the recursive function	software implementation test and checks
	Bitfinex	$72 million	wrong implementation of multi-sig wallet, using hot wallet	
Code Vulnerability	user "devops199"	$159 million	self destruction, accidentall kill of a contract	preventing self destruction and software check
DDoS	BIPS	$1 million	nature of web and network	web application firewall, DDoS protection service, robust network and server system
	Bitstamp	service unavailability		
	Bitfinex	service unavailability		

	Bittrex	service unavailability		
	Bitcoin Gold	service unavailability		
Phishing Attack	Bitstamp	$5 million	uneducated and careless workers, hot wallet usage	instrusion detection and prevention systems, storing in cold wallet, educated carefull users, user credential definion
Man In the Middle Attacks	Ledger Nano	PIN, Security Card Key	Plaintext PIN transaction,	Data Encryption during transaction
Hardware Attacks	Ledger Nano S	Code integrity	non-secure processor	using secure processor and checking software, firmware integrity, protecting sensitive data against side channel analysis, obfuscating operations and transactions, scscall parameter check, zeroisation of flash and RAM
	Trezor	cryptographic keys	no resistance against side channel attacks	
	Keepkey	cryptographic keys	no precaution for timing attack	
	Ledger Nano S	private data, software integrity	no isolation, not cleaning flash after reset	
	Ledger Nano S	Hardware Integrity	not having tamper resistance case, open debug port	
	Bitfi	extract secret phrase and salt	not cleaning memory after root	

5.2 Comparative Analysis of Hardware Wallets

Most known hardware wallets are Trezor One, Trezor T, Ledger Nano S, Ledger Nano X, Ledger Blue, Keepkey, BitBox, BC Vault and Coolwallet S. Each one of them have different specifications and design. Table 5.2 represents the specifications of these wallets according to physical and security aspects.

Table 5.2: Specification of Hardware Wallets

Wallet	**Display**	**Connection**	**Case**	**Protection**	**Pinpad**
Trezor One	128x64 pixels	USB	Plastic	-	2 buttons
Trezor T	Color Touchscreen	USB	Plastic	-	touchscreen
Ledger Nano S	250x30 pixels	USB	steel, plastic	Secure IC, tamper proof	2 buttons
Ledger Nano X	Monochrome	Bluetooth	Steel, plastic	Secure IC	2 buttons
Ledger Blue	Touchscreen	USB	zamak, plastic	secure IC, tamper proof	touchscreen
Keepkey	256x64 pixels	USB	Aluminum	-	one button
BitBox	Led Indicator	USB	plastic		one touch button
BC Vault	128x64 pixels	USB	plastic	-	4 way control pad
Coolwallet S	Monochrome	NFC, bluetooth	plastic	Secure IC, tamper proof	one button

5.2.1 Trezor

There is not much difference between Trezor T and Trezor One according to information provided by Trezor team. Trezor products verify firmware signature before installation and boot. If device detects new firmware is not signed by producer it warns users [85]. This specification corresponds to the FAU_ARP.1, FDP_DAU.1, FDP_RIP.1, FPT_FLS.1 and FPT_TST.1 SFRs. Similarly, secure update procedure

provides FDP_RIP.1 functionality which means that memory is erased on firmware update by bootloader and only genuine firmware can restore back. It is stated that bootloader is write protected.

Allowance of private and public key operations only after user authentication means that FIA_UAU.2 is provided.

Ultrasound hardware seal makes the case hard to open and remain proof of tampering but the plastic case could be considered as low security. Resistance to the physical attacks by ultrasound seal is covered by FPT_PHP.3 SFR, but if there is no security response mechanism it means that this SFR is not fully represented. Physical protection consist of not only invasive attacks but also non-invasive and semi-invasive attacks. In [68], there is information about countermeasures against side channel attacks by using constant time. It should be noted that there are other attacks such as power and electromagnetic side channel attacks, fault injection attacks etc.

In [68], it is stated that authentication security of Trezor is provided by PIN up to 9 digit numbers. Quality metric of authentication mechanism can be defined in FIA_SOS.1. It is also stated that waiting time between wrong attempts increases by a power of two and if 15 unsuccessful attempts are reached, device wipes sensitive data automatically. This mechanism is related to authentication failure handling SFR which is FIA_AFL.1. FDP_ACC.2 access control SFR is also covered by authentication functionality.

Reflashing a Trezor device is possible but this attack will cause wiping device storage and giving warning on every start. Against malicious firmware installation, Trezor devices check integrity and authenticity of firmware [68]. This mechanism represents that data protection SFRs are applied such as FAU_ARP.1, FDP_ITT.3, FDP_RIP.1, FDP_SDI.2, FPT_ITT.3 and FPT_TST.1. There are no secure chips inside of the Trezor wallets but they prefer open source software, firmware verification and write-protected bootloader instead.

Stealing and replacing Trezor wallets with a fake one is prevented by custom home screen specification. Trezor lets users to choose a unique picture for home screen so that fake one could be identified easily. FTA_TAB.1 SFR covers this functionality.

FCO_NRO.2 and FCO_NRR.2 proof of origin and receipt SFRs have to be implemented by the nature of the Bitcoin. We assume that audit data is recorded and reviewed so that

FAU_GEN.1 and FAU_SAR.1 SFRs are implemented. Since every wallet has crypto key generation, deletion and operation functionality, FCS_CKM.1, FCS_CKM.4 and FCS_COP.1 SFRs are applied. Tolerance limit for any fault during the operations are defined in FRU_FLT.2.

FIGURE 5.1: Trezor One Hardware Wallet [5]

5.2.2 Ledger

The most obvious security feature of ledger is that having a secure element. Secure chip which is called as secure element provides protection of cryptographic operations and sensitive data on a high level. Besides isolating external attacks, it mitigates vulnerabilities. This element covers most of the physical functional requirements such as emanation security, resistance to physical attacks, random number generation and stored data protection automatically [6]. Reliability of this element comes from third party independent evaluation against security standards such as CC. Since Ledger's secure element is certified according to CC above EAL5 level, most of the SFRs defined in Chapter 3 are already covered, some of them are FAU_ARP.1, FDP_IFC.1, FDP_ITT.1, FDP_ITT.3, FDP_RIP.1, FDP_SDI.2, FDP_UCT.1, FPT_ITT.1, FPT_PHP.2, FPT_PHP.3, FPT_TST.1, FPT_EMS.1, FCS_RNG.1, FRU_FLT.2, FPT_FLS.1, FAU_SAR.1, FDP_SDC.1, FAU_SAS.1, FCS_RNG.1.

As in Trezor, Ledger wallets provide 8 digit PIN authentication. They have both plastic and metal enclosure. Ledger wallets reset to factory settings and erase private keys when three incorrect PIN attempts reached. Ledger also generated custom operating system named BOLOS. Ledger states that each time device powered on, a mechanism checks integrity of software. Ledger Blue has touchscreen and it provides protected authentication feedback means that no show of PIN characters on screen. These specifications show that some other SFRs are provided by Ledger wallets.

FIGURE 5.2: Ledger Hardware Wallet [6]

5.2.3 Keepkey

Keepkey wallet has a combination of up to 9 numbers PIN protection functionality. The numbers on wallet screen are randomized and they are shown to let the user know where to click on client side browser software extension. There is no button or touchscreen on Keepkey. It enforces waiting period after three unsuccessful attempts. The duration starts at 8 seconds and time continues to double after each wrong attempt. If there is a malware on PC, it cannot obtain the PIN because the order of the digits are scrambled. But if there is a shoulder surfer, he can see the PIN characters easily so we can say that FIA_UAU.7 protected authentication feedback SFR is not applied. Wallet owners must be warned to keep wallet isolated so that nobody can see the digits. Recovery seed function is also supported which consist of 12 to 24 words. In [86], Keepkey describes how a wallet owner obtain logs, so we see that audit records are generated and stored but there is no restriction for audit review. Keepkey's display lets users to verify addresses for safe transaction authorization [7]. Also transactions are signed on the device. It is stated that Keepkey device is not affected by running malicious code on hardware because there is single privileged process and signature control [87]. These specifications show that some SFRs are already implemented such

as FDP_ACC.2, FIA_AFL.1, FDP_DAU.1, FAU_GEN.1, FAU_SAR.1, FIA_SOS.1, FIA_UAU.2, FPT_PHP.3, FPT_RCV.1, FPT_RPL.1, FPT_TST.1.

FIGURE 5.3: Keepkey Hardware Wallet [7]

5.2.4 Bitbox

Bitbox cryptocurrency wallet lack of a display so that there is no way of transaction address confirmation through the hardware wallet screen. Replay detection which is provided by FPT_RPL.1 SFR is not provided and it is hard to understand if the clipboard is changed. It supports that displaying transaction info on a smart phone for users to approve or reject the transaction. Bitbox uses encryption on communication so as to maintain confidentiality and this provides FCS_CKM.1, FCS_COP.1 and FDP_UCT.1 SFRs. But encryption keys are produced and stored on a normal chip instead of a secure chip. There are attacks that reveals security breaches of insecure software implementation [88].

As stated in official developer website [89], Bitbox has true random number generator, it lets only signed firmware to be installed on device and 2nd factor authentication is supported with a mobile phone over encrypted USB communication. Micro SD lets users easy backup the wallet. FCS_RNG.1, FIA_UAU.5 and FPT_RCV.1 SFRs are

related to these specifications. According to Bitbox user guide, after 15 failed password attempts, wallet will reset and delete all information [90].

FIGURE 5.4: Bitbox [8]

5.2.5 BC Vault

BC Vault, relatively new Slovenian product, claims that it offers secure backups while other wallets do not and also claims that it is the most secure cryptocurrency wallet [9]. It should be kept in mind that other manufacturers make the same claim. BC Vault can backup private keys as encrypted on micro SD card or as QR code on papers. Wallet screen provides the transaction details so that sender can check the information. Tamper evident enclosure is used to mitigate physical attacks [9]. BC Vault supports PIN entrance on the device and passwords on the desktop application. Authentication mechanism requires at least one of PIN or password. While Ledger and Trezor are providing hierarchical deterministic key generation which means all keys are generated from a master key, a seed, BC Vault generates keys separated from each other. Firmware upgrade is done after the certificate is signed successfully. It is stated that the application on the desktop also checks the validity of the transaction signed by device by comparing it to the original request made on desktop [5]. When we collect all these features we can say that BC Vault provides FCS_CKM.1, FCS_COP.1, FDP_ACC.2, FDP_IFC.1, FDP_ITT.3, FDP_UCT.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.5, FPT_PHP.3, FPT_RCV.1, FPT_RPL.1, FPT_TST.1 and FCS_RNG.1 SFRs.

FIGURE 5.5: BC Vault and Ledger Nano S [9]

5.2.6 Coolwallet S

Another new hardware wallet is Coolwallet S which is using secure element and having tamper proof design. It provides 2 factor authentication. Recovery seed of 12 to 24 words are created during initialization of wallet and all following keys are derived from this seed. It supports biometric authentication and the display on the device lets users to see and confirm the transaction information. The communication between wallet and mobile app via Bluetooth is encrypted [91]. Like the other wallets using secure chips, most of the security could easily be attributed to the chip. We accept that the same SFRs are implemented as in other wallets which are using secure chips. It is also mentioned in the user manual that address verification is done with Metacert, a protocol protecting the recipient address [10]. Coolwallet S can be used with a smartphone while sending Bitcoin, user need to press the button on wallet, then check the amount and address on wallet display and authenticate on smartphone screen. In this way multi factor authentication can be achieved.

FIGURE 5.6: Coolwallet S [10]

5.3 Analysis and Comparison

In the following table, it will be shown that which security functional requirements are implemented by which hardware wallet. The meanings of the signs in the table are as follows:

- ✓: Device implements the defined SFR
- ? : There is no information about or we could not find whether the device implements or not
- ✯: There is no information but due to the nature of the blockchain and coin system these functionalities are required and accepted as implemented
- ✗: As our best knowledge this SFR is not implemented, device does not have this functionality

Note: Table 5.3 show best of our knowledge and based on public information available. The table cannot be referenced before SFRs are officially verified.

Table 5.3: Comparison of the realization of the SFRs by wallets

SFR	**Trezor One**	**Trezor T**	**Ledger Nano S**	**Ledger Nano X**	**Ledger Blue**	**Keepkey**	**BitBox**	**BC Vault**	**Coolwallet S**	**Bulutwallet**
FAU_ARP.1	?	?	?	?	?	?	✗	?	?	✓
FAU_GEN.1	✯	✯	✯	✯	✯	✓	✯	✯	✯	✓
FAU_SAR.1	✯	✯	✯	✯	✯	✓	✯	✯	✯	✓
FAU_SAR.2	?	?	?	?	?	?	?	?	?	✓
FAU_STG.1	?	?	?	?	?	?	?	?	?	✓
FCO_NRO.2	✯	✯	✯	✯	✯	✯	✯	✯	✯	✓
FCO_NRR.2	✯	✯	✯	✯	✯	✯	✯	✯	✯	✓
FCS_CKM.1	✯	✯	✯	✯	✯	✯	✓	✓	✯	✓
FCS_CKM.4	✯	✯	✯	✯	✯	✯	✯	✯	✯	✓
FCS_COP.1	✯	✯	✯	✯	✯	✯	✓	✓	✯	✓
FDP_ACC.2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FDP_DAU.1	✓	✓	✯	✯	✯	✓	✯	✯	✯	✓
FDP_IFC.1	?	?	✓	✓	✓	?	?	✓	?	✓
FDP_ITT.1	?	?	✓	✓	✓	?	?	?	?	✓
FDP_ITT.3	✓	✓	✯	✯	✯	✯	✯	✓	✓	✓
FDP_RIP.1	✓	✓	✓	✓	✓	?	?	?	✓	✓
FDP_SDI.2	✓	✓	✓	✓	✓	?	?	?	✓	✓
FDP_UCT.1	?	?	?	?	?	?	✓	✓	?	✓
FDP_UIT.1	?	?	?	?	?	?	?	?	?	✓
FIA_AFL.1	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
FIA_SOS.1	✓	✓	✯	✯	✯	✓	✯	✓	✯	✓
FIA_UAU.2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FIA_UAU.5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FIA_UAU.6	?	?	?	?	?	?	?	?	?	✓

FIA_UAU.7	?	?	?	?	?	?	?	?	?	✓
FPT_FLS.1	✓	✓	✓	✓	✓	✯	✯	✓	✓	✓
FPT_ITT.1	?	?	✓	✓	✓	?	?	?	✓	✓
FPT_ITT.3	✓	✓	?	?	?	?	?	?	?	✓
FPT_PHP.2	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
FPT_PHP.3	✓*	✓*	✗	✗	✗	✓*	✗	✓*	✗	✓
FPT_RCV.1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FPT_RPL.1	?	?	✓	✓	✓	✓	✗	✓	✓	✓
FPT_STM.1	✯	✯	✯	✯	✯	✯	✯	✯	✯	✓
FPT_TST.1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FRU_FLT.2	?	?	✓	✓	✓	?	?	?	✓	✓
FTA_SSL.3	?	?	?	?	?	?	?	?	?	✓
FTA_SSL.4	✯	✯	✯	✯	✯	✯	✯	✯	✯	✓
FTA_TAB.1	?	?	?	?	?	?	?	?	?	✓
FTA_TAH.1	?	?	?	?	?	?	?	?	?	✓
FTP_ITC.1	✯	✯	✯	✯	✯	✯	✯	✯	✯	✓
FTP_TRP.1	✯	✯	✓	✓	✓	✯	✯	✯	✯	✓
FPT_EMS.1	✗	✗	✯	✯	✯	✗	✗	✗	✯	✓
FCS_RNG.1	✯	✯	✓	✓	✓	✯	✓	✓	✓	✓

* states that even though the functionality is claimed to be applied, we consider that it is not met at expected level.

The research we have done so far has shown that in order to create a secure product it is necessary to treat the device as a whole system and identify security problems accordingly. When we examine each hardware wallet, we see that each one has weaknesses at different points in their designs. Some of them has weak physical design while some others have no software countermeasures. Almost all of the hardware wallets are attacked and discovered that each one of them is vulnerable to different attacks. Table 5.3, except for the wallet we offer, shows that all the wallets are missing some of the features that might be related to safety. Thus, as we have already mentioned in the previous sections, even hardware wallets, claimed to be the most secure wallets, have been attacked and data leakages occurred. The comparison reveals that using evaluated secure chips to

protect assets in a wallet is the reliable way and lack of secure chip is very difficult to fill with software precautions.

Chapter 6

Conclusion

Hardware cryptocurrency wallet presented in this thesis has more security characteristics and useful design than any other wallet ever made. Comparative analysis shows that the wallet we have developed has covered all possible aspects of security vulnerabilities, while other wallets have deficiencies in certain aspects. Proper design allows this device to be used not only for the cryptocurrency purposes, but also for other ways such as password management, personal identity etc. Also, the presented work is the first study in the field of Common Criteria for cryptocurrency wallets and also hardware wallets. As the security evaluation of the cryptocurrency wallets will be made inevitably, our work will be the most important resource if the CC is used as an official evaluation methodology. Recently experienced vulnerabilities and attacks showed that the lack of evaluation in the test and evaluation leads to huge losses. While researches on blockchain pursued on alternative cryptocurrency solutions and their further applications, not enough effort has been put on usability and security certification of wallets. This study will attract developers, users and CC evaluation labs to put more focus on standardized framework for this evolving technology.

As a future work, the feasibility studies of the proposed hardware wallet design are planned to be made. CC evaluation of the wallet in case of a possible production will be the first officially approved certification. We will leave these works for future projects as it requires more effort and extends scope of this thesis. We believe that test and evaluation processes, especially CC framework, would surely contribute to increasing security of cryptocurrency wallets and developing more secure products.

Bibliography

[1] Paper Wallet Guide: How to Protect Your Cryptocurrency, 2017. URL `https://blockgeeks.com/guides/paper-wallet-guide/`. (accessed 16.04.2019).

[2] M. Gentilal, P. Martins, and L. Sousa. Trustzone-backed bitcoin wallet. In *CS2'17 Proceedings of the Fourth Workshop on Cryptography and Security in Computing Systems*, pages 25–28, Jan. 2017. doi: 10.1145/3031836.3031841.

[3] A. Bialas. Common Criteria Related Security Design Patterns for Intelligent Sensors Knowledge Engineering Based Implementation. *Sensors (Basel). 2011*, 11:8085–114, Dec. 2011. doi: 10.3390/s110808085.

[4] Technical Committee ISO/IEC JTC 1 SC 27. *ISO/IEC TR 15446:2017 Information technology - Security techniques - Guidance for the production of protection profiles and security targets*, 2017.

[5] C. Meiterman. Trezor Vs. Ledger Vs. BC Vault Review An Updated Analysis, April 03, 2019. URL `https://usethebitcoin.com/trezor-vs-ledger-an-updated-analysis/`. (accessed 05.05.2019).

[6] A Closer Look Into Ledger Security: the Secure Element, 2019. URL `https://www.ledger.com/2018/12/03/a-closer-look-into-ledger-security-the-secure-element/`. (accessed 18.03.2019).

[7] KeepKey Safe from Man in The Middle (MiTM), 2018. URL `https://info.shapeshift.io/blog/2018/02/06/keepkey-safe-from-man-in-the-middle-mitm/`. (accessed 18.03.2019).

[8] Tech Company News Editorial. Meet The BitBox One Of The Worlds Most Secure Crypto Hardware Wallets, August 27, 2018. URL `http://techcompanynews.com/`

meet-the-bitbox-one-of-the-worlds-most-secure-crypto-hardware-wallets/. (accessed 15.02.2019).

[9] Altcoin Magazine, April 10, 2019. URL https://medium.com/altcoin-magazine/interview-with-cto-of-bc-vault-alen-salamun-a4ce266c60d7. (accessed 08.05.2019).

[10] Coolwallet. CoolWallet S: hardware wallet review v3, 2018. URL https://coolwallet.io/wp-content/uploads/2018/08/CWS_UserManual_v3.pdf. (accessed 05.05.2019).

[11] P. Gomes. Blockchain Technology The Marketing Value of Digital Permanence. White paper, 4A's, Nov. 2017. URL http://aaaa.org/wp-content/uploads/2017/08/Blockchain-2017-Aug-28-Final.pdf.

[12] I. Grigg. Triple Entry Accounting, 2005. URL http://iang.org/papers/triple_entry.html. (accessed 19.05.2019).

[13] J. Dai and M. A. Vasarhelyi. Toward Blockchain-Based Accounting and Assurance. *Journal of Information Systems*, 31:5–21, June 2017.

[14] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. URL http://www.bitcoin.org/bitcoin.pdf. (accessed 10.11.2018).

[15] D. Drescher. *Blockchain Basics A Non-Technical Introduction in 25 Steps*. Apress, 2017. ISBN 978-1-4842-2603-2. doi: 10.1007/978-1-4842-2604-9.

[16] T. Laurence. *Blockchain For Dummies*. John Wiley, 2017. ISBN 978-1-119-36559-4.

[17] M. Swan. *Blockchain: Blueprint For A New Economy*. Oreilly, first edition, 2015. ISBN 978-1-491-92049-1.

[18] I. Bashir. *Mastering Blockchain*. Packt Publishing, first edition, Mar. 2017. ISBN 978-1-78712-544-5.

[19] J. J. Bambara and P. R. Allen. *Blockchain*. McGraw-Hill Education, 2018. ISBN 978-1-26-011586-4.

[20] A. Usta and S. Dogantekin. *Blockchain 101*. Kapital Medya Hizmetleri, 2017. ISBN 978-605-4584-97-0.

[21] V. Morabito. *Business Innovation Through Blockchain*, chapter 4, pages 61–80. Springer, 2017. ISBN 978-3-319-48477-8.

[22] G. Karame and E. Androulaki. *Bitcoin and Blockchain Security.* Artech House, 2016. ISBN 978-1-63081-013-9.

[23] K. Wüst. Security of Blockchain Technology. Master thesis, ETH Zurich Department of Computer Science, 2016. URL `https://doi.org/10.3929/ethz-a-010703416`.

[24] K. Wüst and A. Gervais. Do you need a Blockchain. *IACR Cryptology ePrint Archive*, page 375, 2017. URL `http://eprint.iacr.org/2017/375`.

[25] I. Lim, Y. Kim, J. Lee, J. Lee, H. Nam-Gung, and J. Lee. The Analysis and Countermeasures on Security Breach of Bitcoin. In B. Murgante, S. Misra, A. A. C. Rocha, C. Torre, J. G. Rocha, M. I. Falcão, D. Taniar, B. O. Apduhan, and O. Gervasi, editors, *Computational Science and Its Applications – ICCSA 2014*, pages 720–732. Springer International Publishing, 2014. ISBN 978-3-319-09147-1.

[26] T. Bamert, C. Decker, R. Wattenhofer, and S. Welten. BlueWallet: The Secure Bitcoin Wallet. In *Security and Trust Management - 2014*, volume 8743, pages 65–80, Sep. 2014. doi: 10.1007/978-3-319-11851-2_5.

[27] D. Efanov and P. Roschin. The All-Pervasiveness of the Blockchain Technology. In *8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2017*, pages 116–121, 2017.

[28] M. Conti, S. Kumar, C. Lal, and S. Ruj. A Survey on Security and Privacy Issues of Bitcoin. *IEEE Communications Surveys Tutorials*, 20(4):3416–3452, 2018.

[29] A. Gkaniatsou, M. Arapinis, and A. Kiayias. Low-Level Attacks in Bitcoin Wallets. In *Information Security, 20th International Conference, ISC 2017, Ho Chi Minh City, Vietnam, November 22-24, 2017, Proceedings*, pages 233–253, Nov. 2017. ISBN 978-3-319-69658-4. doi: 10.1007/978-3-319-69659-1_13.

[30] M. Arapinis, A. Gkaniatsou, and D. Karakostas. A Formal Treatment of Hardware Wallets. Feb. 2019.

[31] S. Volotikin. Software attacks on hardware wallets, 2018. URL `https://i.blackhat.com/us-18/Wed-August-8/`

us-18-Volokitin-Software-Attacks-On-Hardware-Wallets.pdf. (accessed 04.05.2019).

[32] J. Dakto, C. Quartier, and K. Belyayev. Breaking Bitcoin Hardware Wallets, 2017. URL https://cryptotronix.files.wordpress.com/2017/10/breaking-bitcoin.pdf. (accessed 06.05.2019).

[33] J. Hoenicke. Extracting the Private Key from a TREZOR, November 01, 2018. URL https://jochen-hoenicke.de/crypto/trezor-power-analysis/. (accessed 01.05.2019).

[34] D. Yaga, P. Mell, N. Roby, and K Scarfone. Blockchain Technology Overview. Internal report, NIST, Oct. 2018. URL https://doi.org/10.6028/NIST.IR.8202.

[35] FINRA. Distributed Ledger Technology: Implications of Blockchain for the Securities Industry. Report, FINRA, Jan. 2017.

[36] Secure Technology Alliance. Blockchain and Smart Card Technology version 1.0. Technical report, Secure Technology Alliance, Mar. 2017. URL https://www.securetechalliance.org/wp-content/uploads/Blockchain-SC-Technology-WP-FINAL-March-2017.pdf.

[37] E. Piscini, D. Dalton, and L. Kehoe. Blockchain & Cyber Security. Technical report, Deloitte, 2017. URL https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology-media-telecommunications/Blockchain-and-Cyber.pdf.

[38] J. Garay, A. Kiayias, and N. Leonardos. The Bitcoin Backbone Protocol: Analysis and Applications. In *Advances in Cryptology - EUROCRYPT 2015 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26 - 30, 2015, Proceedings, Part II*, pages 281–310, Apr. 2015. ISBN 978-3-662-46802-9. doi: 10.1007/978-3-662-46803-6_10.

[39] R. Pass, L. Seeman, and A. Shelat. Analysis of the Blockchain Protocol in Asynchronous Networks. pages 643–673, Apr. 2017. ISBN 978-3-319-56613-9. doi: 10.1007/978-3-319-56614-6_22.

[40] C. Badertscher, U. Maurer, D. Tschudi, and V. Zikas. Bitcoin as a Transaction Ledger: A Composable Treatment. pages 324–356, Jul. 2017. doi: 10.1007/978-3-319-63688-7_11.

[41] N. Atzei, M. Bartoletti, S. Lande, and R. Zunino. A formal model of Bitcoin transactions. In *IACR Cryptology ePrint Archive 2017*, Jan. 2018.

[42] D. Appelbaum and R. A. Nehmer. Designing and Auditing Accounting Systems Based on Blockchain and Distributed Ledger Principles. Presented at 40th World Continuous Auditing & Reporting Symposium - Newark, NJ, 2017.

[43] G. Hileman and M. Rauchs. Global Blockchain Benchmarking Study. Technical report, University of Cambridge Judge Business School, 2017. URL `https://www.jbs.cam.ac.uk/fileadmin/user_upload/research/centres/alternative-finance/downloads/2017-09-27-ccaf-globalbchain.pdf`.

[44] BitFury Group in collaboration with Jeff Garzik. Public Versus Private Blockchains v1.0. White paper, Bitfury Group Limited, October 20, 2015.

[45] Cryptocurrency Wallet Guide: A Step-By-Step Tutorial, 2017. URL `https://blockgeeks.com/guides/cryptocurrency-wallet-guide/`. (accessed 16.04.2019).

[46] A. M. Antonopoulos. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. O'Reilly Media, December 27, 2014. ISBN 978-1449374044.

[47] M. Palatinus, P. Rusnak, A. Voisine, and S. Bowe. *BIP-0039: Mnemonic code for generating deterministic keys*, September 10, 2013. URL `https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki`. (accessed 11.01.2019).

[48] Deterministic wallet, February 20, 2019. URL `https://en.bitcoin.it/wiki/Deterministic_wallet`. (accessed 11.03.2019).

[49] R. Sachova, M. M. Marcos, and S. H. Revetti. Security of Mobile Payments and Digital Wallets. Technical report, European Union Agency For Network and Information Security, Dec. 2016.

[50] D. Dimov and R. Juzenaite. Security vulnerabilities of cryptocurrency exchanges, June 26, 2018. URL `https://resources.infosecinstitute.com/security-vulnerabilities-of-cryptocurrency-exchanges/`. (accessed 02.02.2019).

[51] Developers guide:Cryptography. URL `https://wiki.trezor.io/Developers_guide:Cryptography`. (accessed 10.03.2019).

[52] H. Lee, S. Jeong, J. Shin, and K. Kou. Common Criteria Requirement of Data Leakage Protection System. *Journal of Security Engineering*, 11:65–78, Feb. 2014. doi: 10.14257/jse.2014.02.10.

[53] A. Mehmood. How Common Criteria Helps Organizations Choose the Right HSM, October 19, 2018. URL `https://www.cryptomathic.com/news-events/blog/common-criteria-helping-to-choose-the-right-hsm`. (accessed 14.03.2019).

[54] Common Criteria Development Board. *Common Criteria for Information Technology Security Evaluation Part 2*, Apr. 2017.

[55] T. Takebe. Trend in Security Evaluation and Accreditation. In *2008 SICE Annual Conference*, pages 1482–1486, Aug. 2008. doi: 10.1109/SICE.2008.4654893.

[56] Common Criteria Development Board. *Common Criteria for Information Technology Security Evaluation Part 3*, Apr. 2017.

[57] Common Criteria Development Board. *Common Criteria for Information Technology Security Evaluation Part 1*, Apr 2017.

[58] Common Criteria Development Board. *Common Methodology for Information Technology Security Evaluation: Evaluation methodology*, Apr 2017.

[59] R. Richards, D. Greve, M. Wilding, and W. M. Vanfleet. The Common Criteria, Formal Methods and ACL2. 2004.

[60] Trusted Computing Group. *Trusted Computing Group Protection Profile PC Client Specific Trusted Platform Module TPM Family 1.2; Level 2 Revision 116*, July 14, 2014. URL `https://trustedcomputinggroup.org/wp-content/uploads/PC_Client_TPM_PP_1.3_for_TPM_1.2_Level_2_V116.pdf`.

[61] G. Shnurenko. How to Hack Bitcoin: All Possible Ways, February 27, 2019. URL `https://u.today/how-to-hack-bitcoin-all-possible-ways`. (accessed 12.04.2019).

[62] C. H. Kateraas. Threats to Bitcoin Software. Master thesis, Norwegian University of Science and Technology Department of Computer and Information Science, May

2014. URL https://brage.bibsys.no/xmlui/bitstream/handle/11250/253952/762896/_FULLTEXT01.pdf?sequence=1.

[63] R. V. Deshmukh and K. Devadkar. Understanding DDoS Attack & its Effect in Cloud Environment. *Procedia Computer Science*, 49:202–210, Dec. 2015. doi: 10.1016/j.procs.2015.04.245.

[64] L. King. Bitcoin Hit By Massive DDoS Attack As Tensions Rise, February 12, 2014. URL https://www.forbes.com/sites/leoking/2014/02/12/bitcoin-hit-by-massive-ddos-attack-as-tensions-rise/#6797db37246a. (accessed 02.02.2019).

[65] Full Drive Encryption international Technical Community. *collaborative Protection Profile for Full Drive Encryption Authorization Acquisition*, February 1, 2019.

[66] F. X. Standaert. *Introduction to Side-Channel Attacks*, pages 27–42. Springer Science+Business Media, 2010. doi: 10.1007/978-0-387-71829-32.

[67] BSI Team. *Security IC Platform Protection Profile with Augmentation Packages BSI-CC-PP-0084-2014*. Bundesamt fÃ¼r Sicherheit in der Informations technik (BSI), 2014.

[68] SatoshiLabs. Security:Threats, March 14, 2019. URL https://wiki.trezor.io/Security:Threats#Hacking_SatoshiL. (accessed 02.02.2019).

[69] J. Wieczner. Hackers Stole $50 Million in Cryptocurrency Using Poison Google Ads, February 14, 2018. URL http://fortune.com/2018/02/14/bitcoin-cryptocurrency-blockchain-wallet-hack/. (accessed 14.03.2019).

[70] L. Batina, N. Mentens, and I. Verbauwhede. Side-channel issues for designing secure hardware implementations. pages 118– 121, Aug. 2005. ISBN 0-7695-2406-0. doi: 10.1109/IOLTS.2005.64.

[71] SatoshiLabs. User manual:Security best practices. URL https://wiki.trezor.io/User_manual:Security_best_practices. (accessed 18.04.2019).

[72] R. Latifa, K. Ahemed, G. Mohamed, and A. Omar. Blockchain: Bitcoin Wallet Cryptography Security, Challenges and Countermeasures. *Journal of Internet Banking and Commerce*, 22(3), Dec. 2017.

[73] M. A. Balanza. Trojan.Coinbitclip, February 02, 2016. URL https://www.symantec.com/security-center/writeup/2016-020216-4204-99. (accessed 10.04.2019).

[74] S. Eshkandary, D. Barrera, E. Stobert, and J. Clark. A First Look at the Usability of Bitcoin Key Management. In *NDSS Symposium 2015*, 2015. URL https://doi.org/10.14722/usec.2015.23015.

[75] 5 High Profile Cryptocurrency Hacks, 2018. URL https://blockgeeks.com/guides/cryptocurrency-hacks/. (accessed 16.04.2019).

[76] B. Curran. The History of The Coincheck Hack: One of The Largest Heists Ever, December 21, 2018. URL https://blockonomi.com/coincheck-hack/. (accessed 11.03.2019).

[77] Imperva Team. Global DDoS Threat Landscape Q3 2017. Technical report, Imperva Incapsula, 2017. URL https://www.imperva.com/resources/reports/2017-q3-ddos-threat-landscape.pdf.

[78] R. McMillan. Want Cheaper Bitcoins? Hit Someone With a DDoS Attack, December 26, 2013. URL https://www.wired.com/2013/11/ddos-bitcoin/. (accessed 08.03.2019).

[79] T. Foltyn. Cryptocurrency exchange Bitfinex plagued by DDoS attacks, December 6, 2017. URL https://www.welivesecurity.com/2017/12/06/cryptocurrency-bitfinex-ddos-attacks/. (accessed 08.03.2019).

[80] E. Kaya. Kripto Para Borsalarına Yönelik Siber Saldırı Vektörleri, February 27, 2019. URL https://medium.com/bitmatrix/kripto-para-borsalarina-yonelik-siber-saldiri-vektorleri-a7abbe8dce5. (accessed 10.05.2019).

[81] C. Guillemet. Firmware 1.4: deep dive into three vulnerabilities which have been fixed, March 20, 2018. URL https://www.ledger.com/2018/03/20/firmware-1-4-deep-dive-security-fixes/. (accessed 19.04.2019).

[82] D. Nedospasov, J. Datko, and T. Roth. Ledger Nano S: Weak Physical Design. URL https://wallet.fail/wallets/nanos/physical-design/. (accessed 19.04.2019).

[83] Z. Whittaker. John McAfee's 'unhackable' Bitfi wallet got hacked again, 2018. URL https://techcrunch.com/2018/08/30/john-mcafees-unhackable-bitfi-wallet-got-hacked-again/. (accessed 02.04.2019).

[84] M. San Pedro. Details about the Side-Channel Attacks on Trezor One Hardware Wallet, March 14, 2019. URL https://medium.com/ledger-on-security-and-blockchain/details-about-\the-side-channel-attacks-on-trezor-one-hardware-wallet-62e2d278e803. (accessed 15.04.2019).

[85] SatoshiLabs. Security measures of Trezor, 2019. URL https://trezor.io/security/. (accessed 18.04.2019).

[86] D. Dapp. How Do I Obtain Logs, December 03, 2018. URL https://keepkey.zendesk.com/hc/en-us/articles/360000217599-How-Do-I-Obtain-Logs-. (accessed 18.04.2019).

[87] KeepKey Users Not Affected: Spectre & Meltdown, 2018. URL https://info.shapeshift.io/blog/2018/01/05/keepkey-users-not-affected-spectre-meltdown/. (accessed 18.04.2019).

[88] S. Rashid. Breaking into the (Digital) BitBox, November 26, 2018. URL https://saleemrashid.com/2018/11/26/breaking-into-bitbox/. (accessed 08.05.2019).

[89] Shift Cryptosecurity. Bitbox, 2018. URL https://shiftcrypto.ch. (accessed 05.05.2019).

[90] M. San Pedro. Digital BitBox User Guide, Mar. 2018. URL https://digitalbitbox.com/download/documents/BitboxUserGuideV1.pdf. (accessed 15.03.2019).

[91] Panama Crypto. CoolWallet S: hardware wallet review, August 3, 2018. URL https://medium.com/@Panama_TJ/coolwallet-s-hardware-wallet-review-c6d397c9c9da.