Safety Analysis "Peer-to-Peer" Electronic Cash System for Bitcoin Platform.

B.A. Tatsuki Monji

Diploma thesis
2019

Tomas Bata University in Zlín
Faculty of Applied Informatics

Tomas Bata University in Zlín
Faculty of Applied Informatics
Academic Year: 2018/2019

MASTER'S THESIS ASSIGNMENT

(PROJECT, ARTWORK, ARTISTIC PERFORMANCE)

Degree, First Name and Surname:	**Tatsuki Monji**
Personal Code:	**A15828**
Degree Programme:	**N3902 Engineering Informatics**
Degree Course:	**Information Technologies**
Thesis Topic:	**Safety Analysis "Peer-to-Peer" Electronic Cash System for Bitcoin Platform**
Thesis Topic in Czech:	**Bezpečnostní analýza elektronického "Peer to Peer" platebního systému platformy Bitcoin**

Thesis Guidelines:

1. **Carry out a literary search for cryptographic payment instruments.**
2. **Describe the technology used by Peer to Peer platform Bitcoin.**
3. **Analyze the current system security and possible security risks of the Bitcoin platform.**
4. **Design a way to safely use Peer to Peer payment tools.**
5. **Assess the benefits of your work and the potential of its use in industry.**

Thesis Extent:

Appendices:

Form of Thesis Elaboration: **tištěná/elektronická**

Bibliography:

asa

1. **BASHIR, Imran. Mastering blockchain: distributed ledgers, decentralization and smart contracts explained. Birmingham: Packt Publishing, 2017. ISBN 978-1-78712-544-5.**
2. **BLANCHARD, Benjamin S. a John BLYLER. System engineering management. Fifth edition. Hoboken, New Jersey: Wiley, 2016. ISBN 9781119047827.**
3. **COMBS, Brett a Tom MITSOFF. Bitcoin decoded. San Bernandino, CA: Propellerhead Marketing Group, c2014. ISBN 978-0-615-95524-7.**
4. **LEE, David a Robert DENG, ed. Handbook of blockchain, digital finance, and inclusion. London: Academic Press, [2018]. ISBN 978-0-12-812282-2.**
5. **SOMMERVILLE, Ian. Software engineering. Tenth edition. Boston: Pearson, [2016]. ISBN isbn-978-0133943030.**
6. **STALLINGS, William. Effective cybersecurity: a guide to using best practices and standards. Indianapolis, IN: Pearson Education, 2018. ISBN 978-0134772806.**

Thesis Supervisor:	**prof. Mgr. Roman Jašek, Ph.D.** Department of Informatics and Artificial Intelligence
Date Assigned:	**3 December 2018**
Thesis Due:	**15 May 2019**

Zlín, 7 December 2018

doc. Mgr. Milan Adámek, Ph.D.
Dean

prof. Mgr. Roman Jašek, Ph.D.
guarantor

I hereby declare that:

- I understand that by submitting my Diploma thesis, I agree to the publication of my work according to Law No. 111/1998, Coll., On Universities and on changes and amendments to other acts (e.g. the Universities Act), as amended by subsequent legislation, without regard to the results of the defence of the thesis.
- I understand that my Diploma Thesis will be stored electronically in the university information system and be made available for on-site inspection, and that a copy of the Diploma/Thesis will be stored in the Reference Library of the Faculty of Applied Informatics, Tomas Bata University in Zlin, and that a copy shall be deposited with my Supervisor.
- I am aware of the fact that my Diploma Thesis is fully covered by Act No. 121/2000 Coll. On Copyright, and Rights Related to Copyright, as amended by some other laws (e.g. the Copyright Act), as amended by subsequent legislation; and especially, by §35, Para. 3.
- I understand that, according to §60, Para. 1 of the Copyright Act, TBU in Zlin has the right to conclude licensing agreements relating to the use of scholastic work within the full extent of §12, Para. 4, of the Copyright Act.
- I understand that, according to §60, Para. 2, and Para. 3, of the Copyright Act, I may use my work - Diploma Thesis, or grant a license for its use, only if permitted by the licensing agreement concluded between myself and Tomas Bata University in Zlin with a view to the fact that Tomas Bata University in Zlín must be compensated for any reasonable contribution to covering such expenses/costs as invested by them in the creation of the thesis (up until the full actual amount) shall also be a subject of this licensing agreement.
- I understand that, should the elaboration of the Diploma Thesis include the use of software provided by Tomas Bata University in Zlin or other such entities strictly for study and research purposes (i.e. only for non-commercial use), the results of my Diploma Thesis cannot be used for commercial purposes.
- I understand that, if the output of my Diploma Thesis is any software product(s), this/these shall equally be considered as part of the thesis, as well as any source codes, or files from which the project is composed. Not submitting any part of this/these component(s) may be a reason for the non-defence of my thesis.

I herewith declare that:

- I have worked on my thesis alone and duly cited any literature I have used. In the case of the publication of the results of my thesis, I shall be listed as co-author.
- That the submitted version of the thesis and its electronic version uploaded to IS/STAG are both identical.

In Zlin; dated:

.....

Student´s Signature

ABSTRACT

Security Analysis "Peer-to-Peer" Electronic Cash System for the master thesis on the Bitcoin platform provides an overview of the security model of peer-to-peer cryptographic currencies. In this master thesis introduces Bitcoin and its design principles, as well as its cryptographic primitives. The focus of the work is analyzing Bitcoin security. It aims to develop a better understanding of how Bitcoin is secured by analyzing various attacking methods related to the system, their relationships and defense mechanisms in Bitcoin system design. We describe that cryptography used in Bitcoin is strong that cannot be breakable, but the system is not tolerant against attacked with computing power and attack by special machines connected to the network. However, lient-side attacks are much more severe because users are highly responsible in Bitcoin. This thesis look into detail of those attacks and explain about defense against them.

ACKNOWLEDGEMENTS

I hereby declare that the print version of my Master's thesis and the electronic version of my thesis deposited in the IS/STAG system are identical.

Foremost, I would like to express my sincere gratitude to my Supervisor Prof. Dr. Roman Jasek (Head of Department, Informatics and Artificial Intelligence, FAI, TBU) for the continuous support of my Master study, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I am very grateful to my classmates Olga Voznyuk and Oni Eniola who were always ready to help, make an interesting and useful conversation or just go and have a rest when it was necessary. Finally, I also would like to thank my family and my friends, who kept believing in me and gave me the courage and support to finish my studies.

CONTENTS

INTRODUCTION

There have been several fiscal crises in recent years with many governments struggling to run their economies efficiently. As a result, financial crises have affected the lives of millions of people. Thus people demand for a new niche currency which is not controlled by governments.

The world has also experienced a rapid development in information technologies. The Internet has evolved into a truly global system that most modern people in developed countries use on a regular basis. The number of services is growing rapidly and the knowledge for the development of complex systems is spreading. This created the opportunity to build a new decentralized digital cash system. It's called Bitcoin. Bitcoin is the world's first and most popular decentralized digital currency [1]. Bitcoin users do not require banks or other central authorities to send and receive money over the Internet. Therefore a user of this system can manage their financial records without needing other parties. Peer-to-peer networking, digital signatures and smart cryptography enables irreversible and fast international payments with low fees.

Since Bitcoin has no central authority, its security lies solely on its system design. However, all systems have their weakness and are targeted when money flows within the system. As Bitcoin is a digital currency, it will take a lot of technical and non-technical attacks. To adopt the system users need to know how safe Bitcoin trading is, what kind of threats they have and how to act to keep their money as safe as possible. This thesis examines the security of Bitcoin, identifies its main weaknesses, and explores ways how threats are or mitigated. The aim of this work is the analysis of various technical attacks on the Bitcoin system, its relationships and the associated defense mechanisms in bitcoin design. Key attacks on the Bitcoin system are presented and analyzed in simple terms thus most of people who interested in Bitcoin security model are able to understand with no need or little extra effort outside understanding this paper. At the same time the biggest security concerns in Bitcoin are discussed in sufficient detail to provide an overview of the attacks, their difficulties and mitigations from different angles.

I. THEORY

1 BITCOIN

Bitcoin is a Peer to Peer decentralized digital currency. It offers fast, secure and irreversible international transactions with very low cost. Not like traditional currency Bitcoin transactions are not handled person-to-person via a banking system. One user sending money to another adds a digital signature to a transaction message stating that another person is now owner of these coins and sends them to the network. Then recipient of this message Send the message to other machines connected to them to spread the information over the Internet.

All Bitcoin transactions are known to users in network, they contain the complete transactions database in theirs Computer and can check transactions by themselves and do not need to trust any other for the information integrity and authenticity. New Bitcoins are created by solving mathematical problem and this process is called mining. First one find the solution for mathematical problem will receive a reward and become the owner of the newly created coins. Mining is not only to create new coins it is also used to add transactions to the database and avoid duplication.

1.1 Bitcoin history

An anonymous entity has registered the domain of Bitcoin.org On August 18, 2008. Later in 31 October 2008, a white paper "Bitcoin: A Peer-to-Peer Electronic Cash System" was published by Satoshi Nakamoto (an anonymous person). To creating Bitcoin Satoshi Nakamoto integrated many existing ideas from the Cypherpunk community such as the digital signatures, proof-of-work system, not rely on third party and Blockchain.

On January 3, 2009, the Genesis Block was created by Nakamoto, the first block of the Bitcoin blockchain. The Genesis block has been hard-coded into the Bitcoin software. Due to the way the code was written, 50 BTC that was created can not be used.

The first Bitcoin transaction occurred between Nakamoto and Hal Finney on January 12, 2009. Nakamoto sent 10 BTC to Finney as test.

On October 5, 2009, the first Bitcoin exchange rate against the dollar was set by the New Liberty Standard. At that time, 1 BTC equal to $0.00076392.

The first transaction to purchase physical goods took place on May 22, 2010. Programmer Laszlo Hanyecz had offered users BTC in exchange for two pizzas on a Bitcointalk.org forum. A teenager named Jeremy Sturdivant accepted the offer. Jeremy sent Hanyecz two pizzas from Papa Johns and received Bitcoin instead.

In 2010, the first Bitcoin exchanges Bitcoin Market and Mt. Gox were established and the first mining pool Slush has successfully mined Bitcoin for the first time. In November, the market cap for Bitcoin exceeded $ 1 million for the first time.

In February 2014, Mt.Gox has been stolen 850,000 BTC cryptocurrency from its customers. This is the biggest theft in Bitcoin history, which was estimated at $ 460,000,000 at the time.

In 2017, the price of Bitcoin exceeded $ 1,000, in June it was over $ 3,000.

On August 1, 2017, part of the Bitcoin community was unable to agree to the proposed changes to the protocol. They support for an increase in block size. Finally they decided to hard fork from the original Bitcoin blockchain and created Bitcoin Cash.

December 17, 2017, Bitcoin price reached a new all-time high of $ 19,783.06 but it decrease so quick to the $13,000 mark by Dec. 31, 2017 and keep decreasing price in 2018.

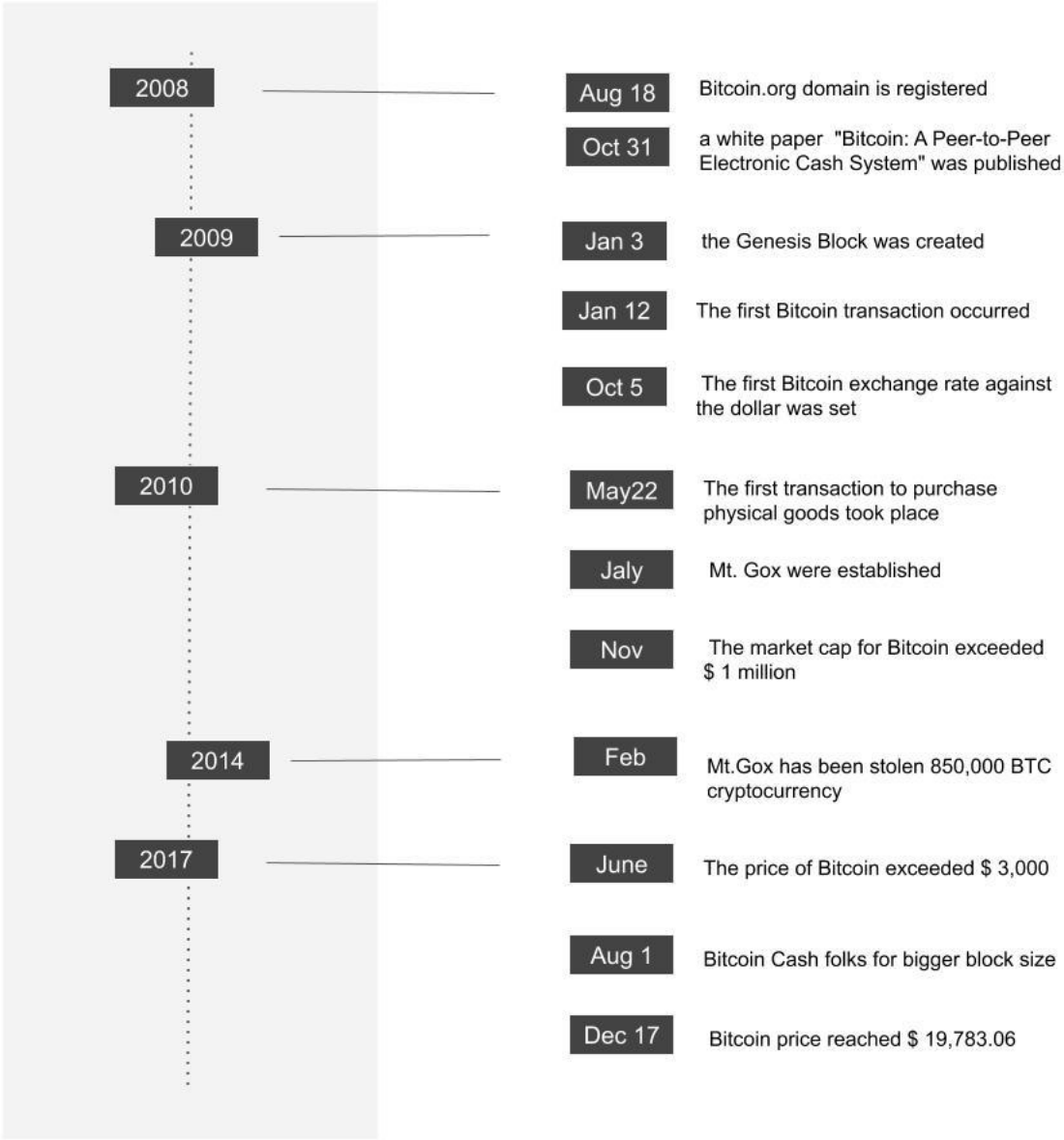

Figure 1: Bitcoin history

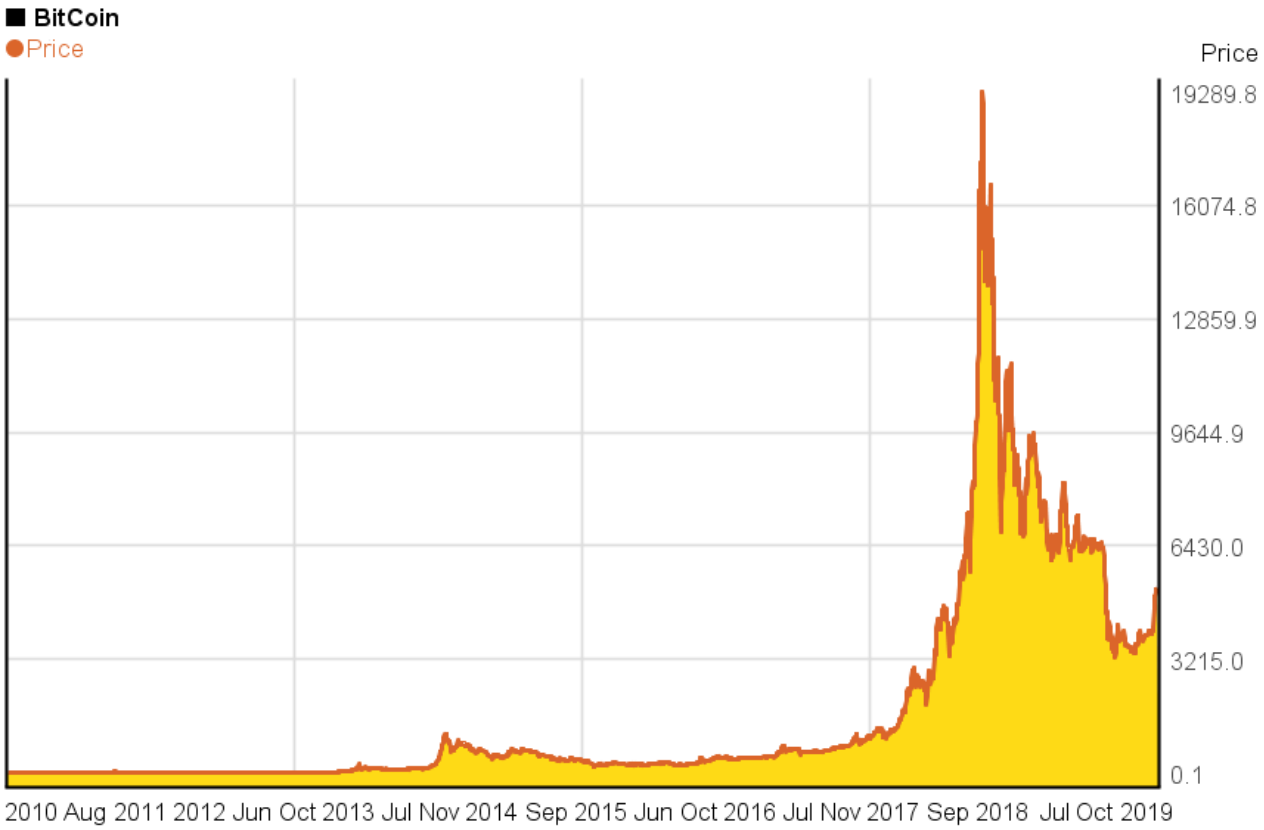

Figure 2: Bitcoin price historical chart

1.2 Decentralization

Not like other virtual currencies, Bitcoin doesn't require any monetary authority or central clearinghouse operated by a company or organization because of its peer-to-peer design. It can be used to purchase products in real world, not only virtual goods and services even though it is not fixed to any real money. Rather than relying on the central bank to empower it to monitor, control and approve transactions and manage the money supply, Bitcoin is supported by a peer-to-peer network similar to file-sharing services such as BitTorrent. [2] . All devices that run Bitcoin software can act as both client and server which can send and receive information about transactions. Participants should share part of their bandwidth and storage space but total setup and operating costs are very low. Owners of Bitcoin wallets can verify all transactions that have ever been made that makes the system very transparent and easily verifiable. Bitcoin transaction fee is optional and amount can be chosen by the payer. However, it is common practice to include a small fee in each transaction, and the original Bitcoin client adds this automatically. Some of the Bitcoin

transactions may have a low monetary value, but process a lot of data. In order to receive a confirmation of such transactions, adding fee to transaction is more than encouraged by the Bitcoin community.

Before transactions are confirmed, its various components are validated like the client's digital signature. After validation, the coins used in the transaction are checked for double spending and the transaction is added to official records, so-called blocks, which are constantly added to each other in a decentralized manner [2]. The transaction messages are sent over the network. Each node is allowed to leave and rejoin the network at any time. In doing so, the longest valid chain of blocks that contained transactions is accepted as evidence while they were leaving network[4]. Due to the fully distributed architecture, we need to expect that most nodes in the network are honest. Majority decision-making mechanism is used to avoid double spending and to resolve any conflicts of interest [2]. In the first place, requiring assumptions seems to be the biggest security issue that Bitcoin has, and there are interesting solutions which enable users to trust the system like paying people for being honest and adding transactions to blocks, verifying them and checking the system. For micro payments low fees are very attractive, where fees would dominate in centralized systems. Bitcoin is also appearing for international sending and receiving of money, as there is no middleman that wants extra money to run its services and therefore incurs no additional fees [2]. With the decentralized design of a digital currency system, we have trading opportunities where we do not have to rely on a government. It enable us to privatize money. We can use it without going through third parties that make the whole process of exchanging value too difficult, time consuming, and high cost. Besides, the money we use today is not designed for the Internet. We have opportunities to deal with the current monetary systems and global networks, but they are not very simple and secure.

1.3 Key and Address

Public key cryptography is the one of the most important factor for Bitcoin. Bitcoin wallet holds private and public key pairs and transactions are done by using those keys. Public keys are used to identify recipients in transaction and private keys sign transaction messages and confirm the currency exchange. The wallet files also store User settings. To reduce the risk of being stolen by hackers it should be encrypted.

A Bitcoin address consist of numbers and uppercase and lowercase letters. Its length is 25-34 characters long and most of the addresses are 33 or 34 characters long. For better readability the address contains neither digit "0" nor the uppercase letter "O" or the lowercase letter "l" or uppercase letter"I" and it usually starts with 1 or 3. A Bitcoin address start with 1 for Pays to PubKey Hash(P2PKH) and 3 for Pays to Script Hash(P2SH).

Figure 3: Examples of Bitcoin address

The Bitcoin address is generated by several steps. The first step is to apply the Elliptic Curve Digital Signature Algorithm to private key to get public key. Then apply SHA-256 to public key and then apply RIPEMD-160. After that calculate checksum of encrypted public key by applying SHA-256 twice and take first 4 bytes of the result. At the last concatenate encrypted public key and the checksum and encode it with Base58 to finally generate Bitcoin address.

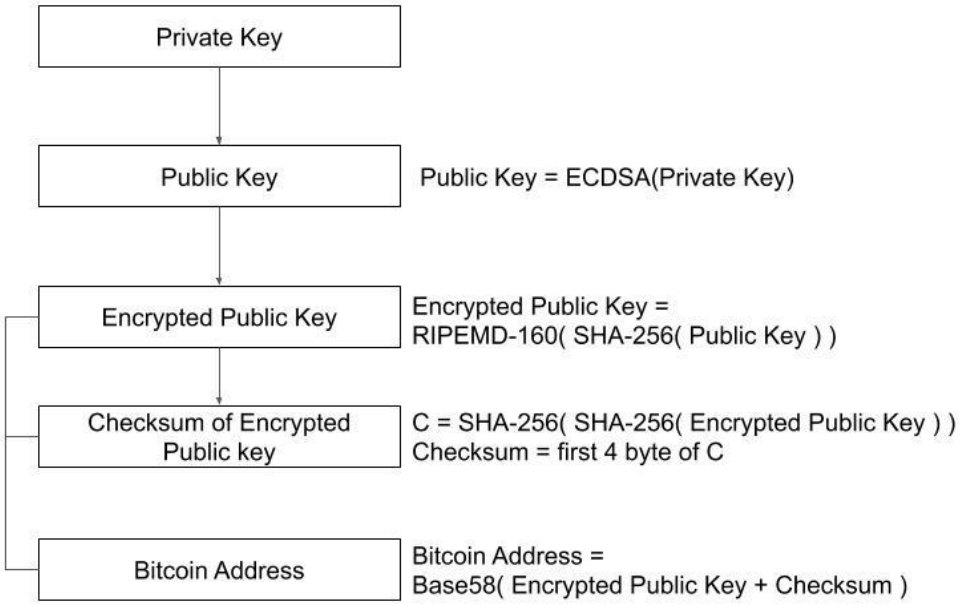

Figure 4: Generating Bitcoin address

The probability of a mistyped address is going to be accepted as valid is about 1 in 4.29 billion which is almost impossible thus it can be ignored[4]. Therefore, protection against typing mistakes in address is very good, although hand typing the addresses for sending bitcoins is probably a rare case.

ECDSA key pairs and Bitcoin addresses are safely created offline using the hashing and encoding rules from the original design and those were not part of Bitcoin network structure. Bitcoin network will knows the address only when the first transaction towards that address is occurred.

Since creating addresses with multiple tools, including the original Bitcoin client, is simple and fast, it's easy for people to use multiple or even thousands of addresses to improve their anonymity. Although, to send Bitcoins to invalid address is impossible but a transaction to an address where the private key is lost is possible and there is no way to use those coins. Those coins are lost forever. That is another reason why it is important for Bitcoin users should keep secure backup for their private key.

1.4 Mining

The process to add transaction records to blockchain is called mining to refers to the searching for gold. A more accurate term, however, would be auditing, since those who contribute are not only finding new blocks, they also check the transactions and help to secure the network. A contributer who has more computing power will have more chances to get new coins as a reward. [7]

A block contains a record of some or all recent transactions, reference to the previous block, a timestamp and an answer to Proof-of-work problem. Proof-of-work is a piece of data which requires costly and time-consuming work to produce but easy to verify and that satisfies certain requirements [6].

Proof-of-work is a nonce in Bitcoin mining. It is an arbitrary number added to the block and hash to a value less than the target value. The level of difficulty of mining is adjusted constantly according to the time required to find the nonces and create new blocks. The goal of the changing it is averaging to create a block over the entire network every 10 minutes [2]. The difficulty raise as the target value become smaller. For this reason, hashes of blocks begin with several consecutive zeroes. More zeros at the beginning of blocks hash as mining difficulties raise. To find a nonce that satisfies certain requirements is computationally intensive. The only way to solve this problem is to try different nonce values. With proof-of-work and mining concepts, possibility of creating counterfeit coins is eliminated. Those counterfeit coins would not be accepted by the network thus those can not be used in transactions.

Mining of bitcoins is a process which need a lot of luck since finding a block is a random event. There is very small chance for solo miners to find a block and redeem a reward since there are thousands of miners compete to get free money. Of course, the idea of free money is flawed because it requires expensive equipments and continuous hashing consume a lot of electricity.

In April 2019 the network hash rate is about 45,000,000 terahashes per second [7]. This is also, in a sense, a network security rate since to attack the network to undo transactions this is the amount of resources that can be considered.

The blocks are connected together in block-chain and each block contains the previous block's hash. Blockchain acts as a database which contains all transactions and all nodes hold it on own computers and keep it up to date by sending and receiving data of transactions and created blocks.

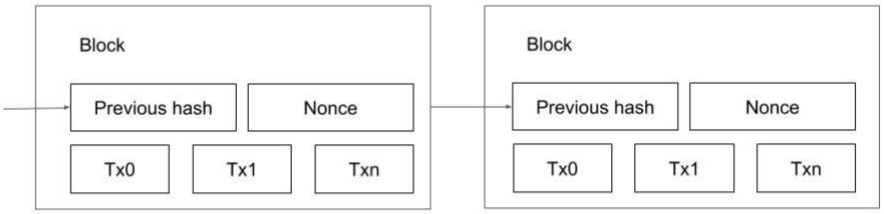

Figure 5:Block-chain

The first block in block-chain is a Genesis block which is hard-coded in the software. There are two reasons for the regular creation of new blocks: One is to ensures to add new transactions to the collectively managed database as quickly as possible and another is to create new coins. This means that mining is a decentralized process that providing incentives to people who contribute their computing power to help keepping the Bitcoin system up and running. The miner receives two types of rewards when he finds an answer to Proof-of-work problem and adds block to the block-chain: One is the transaction fees and another is newly created Bitcoins which depends on the number of blocks in the chain and it is keep decreasing over time. Miners add transaction which send newly created coins towers to miner to the newly discovered block to claim ownership of minted coins.

The original author of the Bitcoin system recognized the potential problem of attacking the network by accumulating more computational power than honest nodes looking

for rewards and described in original Bitcoin paper that as more blocks are added and network grows bigger, difficulty also increases.

To make this attack possible, attacker should be responsible for most of CPU power devoted to hashing in the network to generate the longest chain. This is because attacker who wants to change an earlier block must repeat the proof of the work of the block and any blocks added afterwards before work of the honest nodes combined to network[4]. The nodes in Bitcoin network therefore, in a sense, monitor each other. Since the use of CPU power for mining is rewarded and attacking the network is more harder than playing by the rules, Bitcoin system considered as secure.

There are 2 units for the Bitcoin system, one is BTC and another is satoshi. 1 satoshi equal to 0.00000001 BTC. The reward for mining Bitcoin is steadily decreasing. Currently Bitcoin miners receive 12.5 BTC. The reward for mining new Bitcoin blocks will drop to 6.25 BTC in around one year and will be halved approximately every 4 years until it reach the supply limit. There is a limit for amount of Bitcoin can exist. A maximum amount of Bitcoin can exist is almost 21 million [8]. This is firmly programmed in original Bitcoin software and creates a high confidence of the supply. It is impossible to create extra money out of nothing randomly, need proof-of-work for it. Bitcoins can be divided to eight decimal places, so the range of numbers is very large, which makes it possible to deal with possible deflation due to a smaller supply of money in the future[9].

1.5 Transaction

Bitcoin is described as a chain of digital signatures. The transfer of the coin from one owner to the next owner begins with the digitally signing the new owner's public key and a hash of the previous transaction using the same coins. The signatures are added to the coin and the chain of ownership can be checked by these signatures [4].

The transfer of bitcoins from one owner to another owner occurs on the network by transferring the value received from the transaction inputs to its outputs. A transaction can have multiple inputs and outputs. Special cases of transfers are the transaction of newly minted coins to miner. These are transactions without input [2]. In other cases output of a previous transaction associated with the coin is used as input basically. To make these links visible and verifiable in the block chain, hashes of previous transactions with scripts containing the public key of the receiving party and the sender's digital signature are included in inputs. Bitcoin address or public key's owner are announced as new owners of the Bitcoins and given the right to redeem the outputs and thus use them as inputs to other transactions. The previous owner sign the hash of the transaction with his private key. This helps to prove that the money was actually sent by the actual owner of the address from which the coins came.

An input claims the full value of the previous output. However, an output does not need to claim all bitcoins available from inputs. A portion of the value may not be redeemed and is noted as a transaction fee which is given to the miner generates the block that adds the transaction to the block chain [2]. When the total value of inputs for the transaction exceeds the required amount a special output is created to add a change and keep part of the value to the paying party in the transaction. This works in a similar way to euro payments where an item worth 53€ is bought and paid with a 20€ note and a 50€ note a customer gets back 17€, the difference is that customer hold a Bitcoin which worth 17€ in Bitcoin transaction.

An example transaction is shown in Figure 5. First, the recipient gives his address to the sender and requests 65 BTC for services or some goods. The sender has 3 addresses for which he has private keys and therefore can sign the outputs of previous transactions. Now using them as inputs for this transaction. The outputs that he can claim and transfer to the recipient are amount to a total of 70 BTC. The sender decides to leave the associated fee of 1 BTC and return the remainder of the funds over 65 BTC sent to the recipient to an address

in their own wallet. This results in change from the transaction and there is now another address with coins that they can redeem and use for payments as their wallet also contains the private key for that address.

The transaction is sent to the network and propagated by all nodes in network now. Each node add it to their memory as an unconfirmed transaction until this transaction is added to block and sent to them. A miner who has found a solution to this block collects this 1 BTC fee along with other fees and the block discovery reward. A miner checks the transaction and validates it by adding this transaction to block. The confidence level that the transaction cannot be reversed raises every time other block added to the chain after this block. Of course all this is done by software.

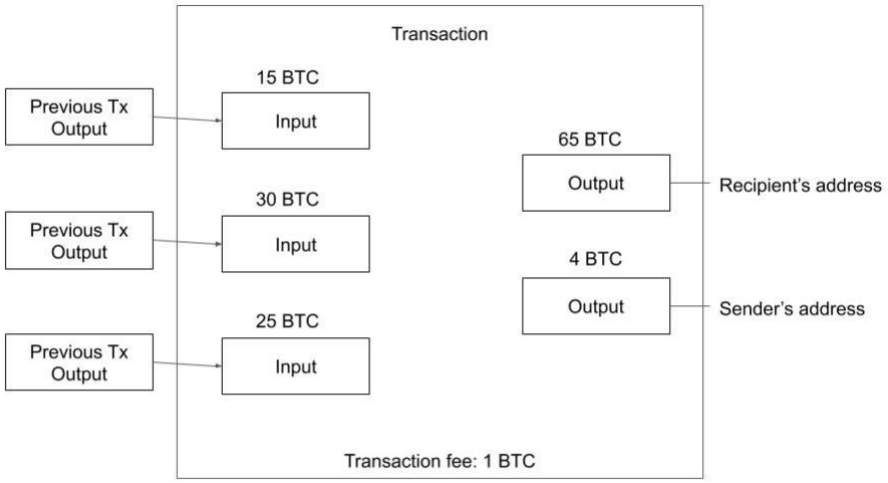

Figure 6: Example transaction

In a transaction its total input value must be greater than its output and the difference going to miner as transaction fees. The coin spender must display the title for each input used in the transaction. This is tested by evaluating the input script, a description of how the owner of the coins can access the coins from outputs of other transactions that are used as inputs[2]. Typically the script displays the public key of the person receiving coins as the address and signature of the person who now wants to spend the coins. The digital signature over the transaction hash proves that the spender has the private key for the key pair with the public key being displayed in scripts of previous transactions output referenced by the transaction. Testing the scripts means checking that all inputs used contain the spender's public key in scripts and the previous owner's signature over the transaction hash is valid.

In digital currency systems Accounting techniques are needed to store and manage rights over time. These concepts help to build complex systems that ensure that no value is lost as long as everyone follows the rules and it must be possible to check where rules are not followed easily[2].

Bitcoin may have demonstrated the first successful implementation of triple-entry bookkeeping on a large scale and therefore Bitcoin has the largest impact on the design of digital currency systems in accounting. In triple-entry bookkeeping, the record of a transaction is debited from an account and credited to another account and there is a cryptographic signature over this transaction, and the third party verify that signature. This simple idea is somewhat revolutionary for accounting and a huge improvement in double-entry bookkeeping that has been used for more than 500 years [10].

In Bitcoin, all nodes in the network hold information about all transactions and miners hash the transactions into new blocks. Transactions are only confirmed after being included in a blockchain and are therefore acknowledged in a collectively maintained timestamped list of all known transactions. The level of confirmation depends on the number of blocks added to the block chain after the transaction, since each one makes less possible that the added block will not be a valid one. This means that both miners and all Bitcoin users are third party in Bitcoins bookkeeping scheme.

Once the miner has added a transaction to a block it becomes more difficult for someone to change it because they need to regenerate all the blocks after the transaction therefore making double spending and reversing previous transactions are practically not possible. This means that Bitcoin transactions become irreversible quickly as new blocks are added to the block chain and no chargebacks [2]. This is one of the benefits of bitcoins over credit cards: merchants have greater assurance that the money they receive is final and no one can cancel the payments. Therefore, the fees are also significantly higher for credit card transfers than for bitcoins. Because credit card fraud is wide-spread, honest customers also pay for fraudulent activity online.

Credit card data can also be lost to hackers who sniff to traffic with transaction information, gain access to databases which hold the numbers or use phishing methods to trick people into providing their credit card information. Bitcoin users do not need to deal with these problems. The private keys are never transmitted over the network, they are not managed on centralized servers, and casual users do not even know how to accidentally hand over their

private keys while knowledgeable users are unlikely to be victims of such an attack. This makes phishing attacks against Bitcoin users useless. Bitcoin users, however, are not completely safe from hackers because there is certain malware target Bitcoin user that tries to get their wallet files with keys to spend their coins.

1.5.1 Transaction format

Transaction is broadcasted to Bitcoin network in a serialized byte format that's not human-readable, called raw format.[2]

```
0100000001678d18a403a92e8f119f92f20e6ecdd5594
a189eb66cb825d09a64749be097eb010000006b48304
5023700d078e825af7688250fcb9fe5c19b1fea64f8901c
b6987050ff278063431ad99d02203de2fa3b5b53fa567c
6896628993411815381 42eeb1395a9e823693b404a9c
8a01210304dfe2d6554170f23c3ced0ea1101f5f306fc53
bb67b40f615a7c4db6c292211ffffffff02c8541700000000
0017a91430897cc6c9d69f6a2c2f1c651d51f22219f1a4f
68758d2620100000000017a914b3379c34267ce2a41c9
6f7a13158f5d9c30c8e5f8700000000
```

Figure 7: Raw transaction data

```
{
  "version": 1,
  "locktime": 0,
  "vin": [
    {
      "txid":
      "7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18",
      "vout": 0,
      "scriptSig" :

"3045022100884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c75c4ae24cb02204b9f0
39ff08df09cbe9f6addac960298cad530a863ea8f53982c09db8f6e3813[ALL]0484ecc0d46f1918b30928f
a0e4ed99f16a0fb4fde0735e7ade8416ab9fe423cc5412336376789d172787ec3457eee41c04f4938de5
cc17b4a10fa336a8d752adf",
        "sequence": 4294967295
    }
  ],
  "vout": [
    {
      "value": 0.01500000,
      "scriptPubKey": "OP_DUP OP_HASH160ab68025513c3dbd2f7b92a94e0581f5d50f654e7
OP_EQUALVERIFY OP_CHECKSIG"
    },
    {
      "value": 0.08450000,
      "scriptPubKey": "OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8
OP_EQUALVERIFY OP_CHECKSIG",
    }
  ]
}
```


Figure 8: Transaction message

Table 1: Transaction message

<table>
<tr><td>version</td><td>version of transaction data structure</td></tr>
<tr><td>locktime</td><td>Time when transaction can be valid</td></tr>
<tr><td>vin</td><td>Inputs<table>
<tr><td>txid</td><td>Transaction ID</td></tr>
<tr><td>vout</td><td>The index of the output being spent within the previous transaction.</td></tr>
<tr><td>scriptSig</td><td>Raw hexadecimal encoding of the script</td></tr>
<tr><td>sequence</td><td>Legacy 4-byte sequence number</td></tr>
</table></td></tr>
<tr><td>vout</td><td>Outputs<table>
<tr><td>value</td><td>Value in this transaction output, in satoshis</td></tr>
<tr><td>ScriptPubKey</td><td>Raw hexadecimal encoding of the encumbrance script for this output.</td></tr>
</table></td></tr>
</table>

1.6 Cryptography

When creating digital cash systems the idea that cryptography solves mathematical problems came up. It offers useful features such as confidentiality, integrity and authenticity. Authenticity and integrity are more important of those for Bitcoin. It must be mathematically verifiable that the sender of the coins is actually someone who has the right to spend them and that his spending messages are not being changed in the network. It is explained in Ian Grigg's 7-layer digital currencies model that cryptography is at the bottom level and is the basis of all these systems. Properties available backed by cryptography are used by disciplines at higher levels of the model to create a secure system [2]. The system crashes if cryptography fails in a major way.

Several security experts, such as Dan Kaminsky, have reviewed the system and code of the original client software and found that use of cryptography in Bitcoin is unorthodox. Kaminsky noted that Bitcoin is really well designed and interesting cryptographic solutions work together to provide security that is sufficient for the peer-to-peer network structure dealing with money [11].

Bitcoin is build on public-key cryptography using the Elliptic Curve Digital Signature Algorithm (ECDSA). To generate Bitcoin addresses and to link both transactions and blocks, well known hashing algorithms SHA 256 and RIPEMD-160 are used. Block-chain is basically a timestamp based hash chain. The way Bitcoin transactions are linked together and added to blocks is similar to a Merkle tree.

ECDSA offers a variant of the digital signature algorithm which uses public-key cryptography based on elliptic curves over finite fields. The algorithm has this name because the curves used to calculate the keys are described by cubic equations that resemble finding the circumference of an ellipse but they do not represent ellipses. The equation for such a curve can be described in a formula.

$y^2 = x^3 + ax + b$ [2]

As the highest component contained therein is 3, this is a cubic equation. To draw such a curve component, y can be computed if you know the component x and specify the values of a and b. For simplicity, we exclude other mathematical parameters, such as

points on the curve and the prime modulus that limit the length of the keys and use only one reference point called G which defines the curve. When G is multiplied by a random number that we use as a private key, we get another point on the curve. Now we use that point as a public key which we can share with others. Hash of the transaction message is signed by private key and that can be verified by checking it to the public key[18].

RSA in the most commonly used digital signature algorithms based on the difficulty of factoring large integers. However, the key lengths for secure RSA have increased, and it is important to keep the size of the messages sent over the network in peer-to-peer systems under control. ECDSA provides the same security for a smaller key size and reduces the processing overhead and amount of data needed to be stored and transfered over network. On the other hand, no mathematical security proofs have been published and they have not been used as algorithms using RSA and therefore the confidence level are not so high [2].

SHA-256 is the most important hashing algorithm used in Bitcoin. The secure hashing algorithm is a cryptographic function which takes a block of data with arbitrary length and always returns a 256-bit string, so finding the message from the digest that is returned after calculations within the function is almost impossible and if one message in data is changed the hash value will most likely change drastically. The hash is also easy to calculate and it is alomost impossible to generate the same hash from two different messages. In fact, no collisions were found for SHA-256 [2].

To find a SHA-256 hash of data is a process that performs bitwise operations on the message that is padded to a length of modulo 512. Operations such as exclusive or and or, bitwise and, logical shift right, bit rotation and integer addition are performed in 64 consecutive rounds modifying the message in 512-bit blocks[12].

Each block has a header in a Bitcoin block-chain. This header contains the version, previous block header hash, Merkle root, timestamp, difficulty of mining and a nonce [13]. If the block header is hashed and the size of result is smaller than the difficulty of mining, a new block is added to the chain and the miner adds that block receives a reward. To find such a hash, the nonces integer value is constantly incremented.

The Merkle root in the header is a cumulative hash of all transactions in a block starting with mining reward sent to the miner address. This means that all miners are trying to find nonces that result in a hash that satisfies the difficulty with different block headers.

While a miner attempts to find the block, new transactions are continually added, as new valid transactions are received from other nodes in the network, therefore changing the Merkle tree and its root hash. During mining the timestamp in the block header is kept up-to-date.

Merkle Root is similar to a summary of all transactions in a block. It is generated by hashing transactions in Merkle Tree. Merkle tree is a binary tree in which the root is found by hashing data by 2 consecutive pieces of information and then doing the same on the next level with the resulting digests from the previous round until 1 cumulative hash is generated. If one of the nodes in the tree has no pairing value, it is concatenated with itself before the tree is expanded, so that there are 2 input hashes for all intermediate nodes. The graphical representation of a Merkle tree of transactions in a block is shown in Figure 8.

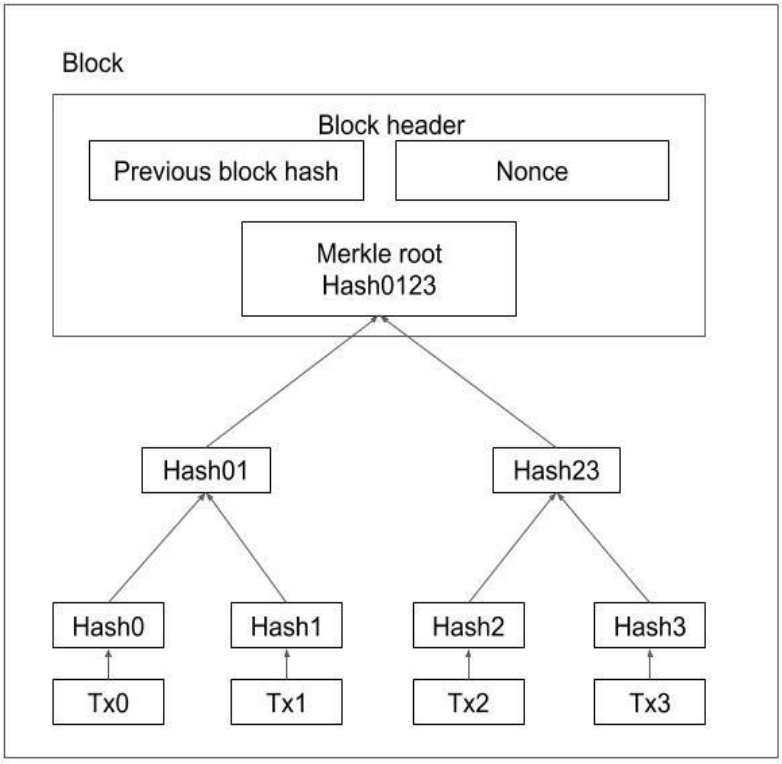

Figure 9: Merkle tree

A timestamped linear chain of blocks, which is known as a block-chain in Bitcoin. In this chain, each block, with the exception of the first block, contains a reference to chronologically previous block and the block header hash is referenced in the next block added after it. Double SHA-256 is used for both mining and building the Merkle tree from the transaction hashes within the block and the result is

interpreted as a little-endian number. Little-endian means that the hash starts with the least significant byte in value. In this way, the mining hashes begin with leading zeros to meet the difficulty of mining.

II. ANALYSIS

2 BITCOIN SECURITY

2.1 Breaking Cryptography

The strength of Bitcoins is that it uses cryptography in a way that no other system has ever done before and that it actually works. It is a currency whose administration does not require a central party. Everything is defined by laws of mathematics. However the cryptographic system can only be as strong as the algorithms it is based on and that the system fails if one of them is broken [18]. This is particularly true for Bitcoin as it is a system based heavily on cryptographic knowledge. Failing of algorithms for Bitcoin would mean that one of important cryptographic systems was broken. Bitcoin use ECDSA, SHA-256 and RIPEMD-160. All of them are published algorithms with a lot of research going into them. Is that possible that someone break cryptographic algorithms and what happens to Bitcoin when one of the important algorithms is broken?

2.1.1 SHA-256 collisions

There are two different kinds of attacks to worry about for SHA-256 or other hashing algorithms: collision and preimage attacks. Collision is a situation in which the same digest value are hashed from different inputs. To find a collision for a SHA-256 via brute-force attack is possible because only a limited number of different hash values can be generated.

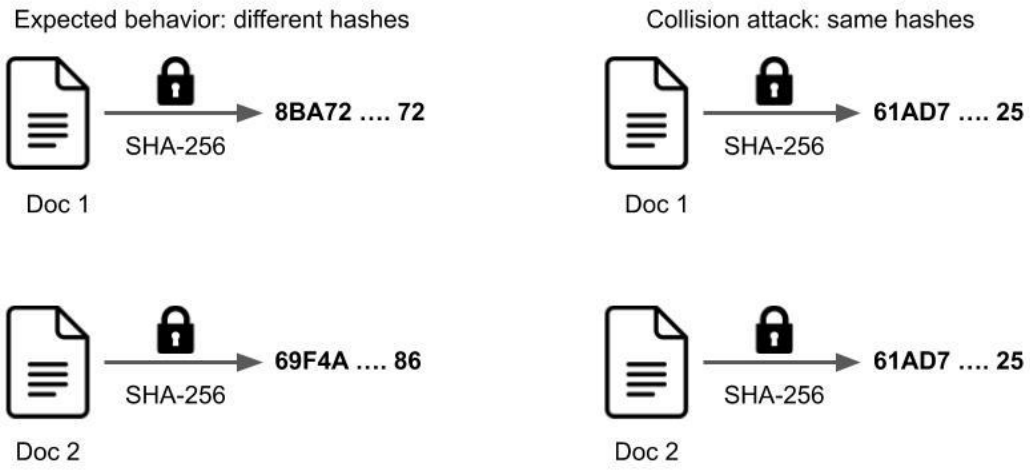

Figure 10: SHA-256 collision

There are a total of 2^{256} hashing results, so it is very unlikely that collisions will occur and we are not worried about such a small possibility. On average, a good attacker who takes advantage of the birthday paradox will likely find a collision in "only" 2^{128} attempts for SHA256, and we need much better chances to find a collision to regard an algorithm as defective. If there is more easier method of finding collisions than brute forcing due to cryptanalysis the algorithm is assumed to have a flaw[14].

Chinese cryptographers broke SHA-1 in 2005. They developed a method to find collisions 2000 times faster than brute-forcing [15]. Their method has since been surpassed by other cryptographers, and machines have become much more powerful in the last 14 years, but to find a collision would still cost a lot of computing resources and luck. If we theoretically envision a crypto-currency system that resembles Bitcoin developed before 2005 and uses SHA-1 as the main hashing algorithm, what would breaking of the algorythm mean to the system, publishing the first paper on how to find collisions faster than brute-forcing 14 years after?

First, Bitcoin would theoretically not be safe if it used SHA-1, but attacks are still not well relate into practice and to find usable holes in a system would not be easy. Hashing is used primarily in mining and transactions in Bitcoin. For transactions, the hash of the transaction

must be signed to transfer the value of the coins to another user. If someone can find a way to create a transaction that has the same hash value as the original one, he would be able to steal the coins by adding himself as the recipient of the coins. The original sender's signature on the transaction hash would be valid, so the transaction would also look like a valid one. However, there are limitations to this clever attack. Attackers need to find this very specific collision, rather than just a collision, a transactional message containing their bitcoin address instead of the intended recipient address would have to have the same hash. For that, the weaknesses found in SHA-1 are far from sufficient. In addition, the attacker must be faster than the owner of the coins to spend them.

After the miner has added the transaction to a block, the attacker can use the outputs of the previous transaction and spend the coins. Both transaction messages refer to the same previous transaction with different scripts added with different owners having the rights to spend the coins using the outputs of the previous transaction. Attacker's efforts become unusable, if he finds the colliding transaction hash after the original owner has already spent the outputs,

Since there is a remote possibility that multiple transactions will hash into the same digest in the future, mitigation is developed well before collisions in SHA-256 make such an attack possible. The default Bitcoin client does not add transactions to the database when they come with a hash already stored there. It takes the incoming transaction as a duplicate and discards it. However, for the protocol in general this can be a problem if some users use thin clients that do not keep all transactions in their database.

To steal the transaction fees and block the discovery bonus, or invalidate some or all transactions for double-spending attacks or denial-of-service, attackers want to find collisions in block hashes . How possible collisions do affect the mining process and the integrity of the block chain? Not like transactions, blocks do not live by their alone. There may be two transactions resulting the same hash value in the block chain and both may reference the same previous transaction due to a collision. However, this is not possible for blocks because they form a timestamped linear hash chain. An attacker's block with the same hash value as one of the previous blocks is not added to the chain if it points to the same previous block as the block it wants to replace. Each newly discovered block has a previous blocks hash and timestamp in the header and the blocks must be in chronological order. The default Bitcoin client does not accept a block with a hash previously stored in the database.

In order to establish a secure timestamping service, the hashing functions used on the server side do not have to be collision and preimage resistant and do not even have to work in one way. This means that breaking the hashing algorithm of SHA-256 does not have real effect in terms of integrity of the block-chain. The old chain would remain unharmed with all transactions hashed there, if a change in the hashing algorithm would be required for other reasons discussed in this chapter. Then, the hashing would continue with the last solved block with old hash as input reference point for the new block, in that block community has agreed upon as the starting point of the new hashing algorithm for mining process.

Another theoretically possible cryptographic attack is to improve the hashing algorithm for SHA-256. If someone could find a way to find double SHA-256 of blocks header significantly faster than others, he would gain huge advantage in mining. They may gain a monopoly in adding blocks to the block chain using a large amount of computational power and undo their own transactions or use them for denial-of-service against miners and regular users by creating empty blocks and not including transactions. The impacts of this will be discussed in the next chapter with more detail. Improving the SHA-256 hashing algorithm might affect Bitcoin only if the improvements remain private. When many miners use more efficient algorithms, the level of mining complexity increases and the system continues to function as usual.

2.1.2 Attacking transaction signature

We also need to investigate possible collisions in RIPEMD-160. This is 2^{96} times more likely to happen than collisions in SHA-256 because the length of hash is 160 bits instead of 256 bits. RIPEMD-160 is used to create Bitcoin addresses which identify to where coins are sent.This means that someone who finds an ECDSA key pair where the public key has the same RIPEMD160 digest value as another person's Bitcoin address can spend all the coins that address holds. However, to create this type of collision, a valid ECDSA key pair that hash into value collides with the RIPEMD-160 hash value would have to be found, and for a public key hashed to an address, first using SHA-256 and then using RIPEMD-160 before computing a double SHA-256 checksum and encoding it to a Bitcoin address[5].

In 2006, a team from Graz University of Technology showed that the methods for detecting collisions in SHA-1 or RIPEMD were not extended to RIPEMD-160 and the algorithm was secure for known attacks[16]. This means that the only method for attacking is brute-forcing, which consist to generate ECDSA key pairs before being hashed with SHA-

256 and RIPEMD-160. Theoretically, the use of RIPEMD-160 makes the Bitcoin protocol less secure because it offers a shortening of public keys that can be conveniently used as addresses because of its shorter hash length. In reality, however, finding collisions requires too many rounds of computation to make such an attack feasible.

Then there are attacks on ECDSA. If someone could find a way to calculate private keys for key pairs where corresponding Bitcoin address holds funds, he could spend it because the private key is all he needs to sign transaction messages and Passing on the value. In Bitcoin private key is a 256-bit integer [2] therefore it has 2^{256} different values and therefore it is more resistant to brute-forcing than the Bitcoin address which is created from the public key by hashing it with RIPEMD-160, which has 2^{160} different values. That means one Bitcoin addresses balance is possible to be redeemed with 2^{96} different key pairs on average. Additional computational issues are discussed in the previous paragraph: One would need to compute a public key for a private key and then hash it twice using two different algorithms.

Let's look at the chances of an attacker trying to find a RIPEMD-160 hash that collides with another Bitcoin address to spend the coins. To find a collision would require at least 2^{80} hashing attempts on average. Suppose an attacker has the same processing power as all miners currently trying to solve a block that is about 14 terahashes (14×10^{12} hashes) per second [7]. From the mining output, total computing power of the Bitcoin system is calculated. Within 1 mining output hash, SHA-256 is computed twice, because in most cases using the algorithm in Bitcoin takes double SHA-256 of the input. Let's generously assume that SHA-256 hashing in the mining process takes as much time as the whole arithmetic difficulty of creating an address from a private key is required for an attacker. We will see that the attacker succeeds on average in $2^{80} \div 14 \times 10^{12}$ seconds, which is more than 2700 years old. We have to keep in mind that computing power increases over time. Let's double it every 18 months, as often quoted in the version of Moore's Law [17]. Now we can find a private key in about 16.25 years, as shown in Appendix A.With highly optimistic estimates, it takes more than 16 years of constant hashing to find a private key from a particular Bitcoin address. Brute-forcing is not feasible from our current knowledge, but we need to keep an eye on developments in cryptography, computation power and perhaps even quantum computing and be able to adapt the algorithms used in the system.

If it is not possible to attack ECDSA keys through brute forcing, we need to find a better way to attack the bitcoin signature algorithm. Bitcoin uses the elliptic curve secp256k1, which has a 256-bit private key and is based on the Koblitz curve [18]. Algorithms that use Koblitz curves are not part of the National Security Agency, ANSI or other standards and therefore are not as extensively studied and analyzed as much as some other ECDSA-s. Therefore, it can be considered less secure. Bitcoin is the only widely used system that uses Koblitz Curve-based ECDSA, and it looks like the author of Bitcoin did not make the best choice, to focus on speed rather than security. At the same time, no vulnerabilities are released for ECDSA and keys are hidden behind hashing algorithms.

Suppose someone has found an actual vulnerability in ECDSA implemented in Bitcoin and was able to crack the algorithm and trivially generate private keys from public keys. Now attackers could forge signatures and sign transaction messages to spend coins that they are not owning. But it would not be possible to get these keys to steal user money for Bitcoin attackers, because they would first need the public keys to begin calculating the private keys of those key pairs. However, the public keys are hashed in the system. A successful preimage attack on both RIPEMD-160 and SHA-256 is required before the vulnerabilities found in ECDSA is used, since the public keys are not broadcast in the network before the coins are signed over to the next owner[19]. For attackers only Bitcoin addresses are available for most addresses and these addresses are created by first hashing the public key with SHA-256 and then with RIPEMD-160. This means that only addresses that are reused are affected by this attack because they have revealed their public keys. However, this is not currently a problem as there are no known vulnerabilities in ECDSA and users can increase their security and anonymity by using different addresses for each transactions.

2.1.3 Preimage attack

A preimage attack on a hash function means to search for the original message from the hash value generated by the hash function. In addition to the mandatory execution of a preimage attack to find private keys, preimage would also help mining coins faster. If they have found a way to generate a nonce from the hashes which satisfies the level of difficulty required for a particular block, they can present it as proof-of-work and collect fees and reward for finding a new block and adding it to the chain. This type of preimage attack would be interesting, as multiple hashes can be attacked and the attacker can also control a part of

the message which is going to be hashed. The Merkle root can be changed by attacker. Attacker can decide which transactions to add to the block that he is trying to hash and which address receive the reward for mining. At the same time, he can only focus on finding the integer value of the nonce.

Currently, the best preimage attack for SHA-256 is against up to 43-step version of the hashing algorithm. The 64-step process is still safe from this meet-in-the-middle attack [20]. Meet-in-the-middle attack means that to attack a hash function's cryptography by working from both ends of the hash at the same time. An attempt is made to take the possible message values closer to the hash digest while bringing hashes closer to the original message until they meet in the middle and detect the input of hashing. In principle, this is exactly the type of attack that could be successful against Bitcoin since the search for a suitable nonce is meeting in the middle. Additional difficulty in attempting a preimage attack is caused by the fact that block headers use double SHA-256 but in the same time a found preimage need not be specific. Any nonce that helps to hash into any acceptable hash values is sufficient. This is one of the attacks that requires further investigation as Bitcoin specific meet-in-the-middle attacks against Double SHA-256 may be possible. If someone has found a method for it, it is quite likely that he will not release them, as even a small advantage in mining is of valuable.

Currently, bitcoin cryptography is very strong: brute forcing is not possible, algorithms are very strong and in the case of algorithm weakening, several mitigations already exist beforehand. However, with cryptanalysis and computational speed developments, longer key sizes and hash lengths or improved algorithms need to be implemented in Bitcoin in the future. Although the author of Bitcoin has mentioned the possibility of changing cryptographic algorithms in the system for users seamlessly, in the unlikely event that SHA-256 will get broken soon[20], there is no concrete plan for doing so.

2.2 Attacking with computing power

Bitcoin fights against double spending by including all broadcast transactions into block chain. Block chain is the database which holds all transactions and the branch of the chain with the highest computational cost associated with it is trusted by nodes in the network [21]. Honest miners build new block on to of the longest valid chain. They will receive reward of adding new block to chain and if they intentionally or accidentally add blocks to the chain that the network does not consider to be the main branch, the coins they receive by claiming the block discovery bonus and transaction fees are no longer available since they are not part of trusted chain. Bitcoin clients should also only trust the transactions included and confirmed by several blocks added to the chain after that, so it is clear that they are part of the main chain and not one of the orphaned chains that are not built on top of the blocks which carry the highest amount of calculations with them.

2.2.1 Double spending attack

The branching of the block-chain may be intentional in the case of an attack, but also accidentally when several new blocks are discovered and sent to the network a few seconds apart. In this case, the nodes in the network that generate the blocks begin to build on top of the block they first received. Now the block to which another new block refers first, becomes part of the main chain and all others remain as orphans, since this branch associated more computational effort [2]. Transactions in orphaned chains return to the unconfirmed state and are added by miners who later build new blocks.

The attacker, who can create a block-chain for which he has a proof-of-work, a level of difficulty matching the hashing speeds, and a higher computational effort than the builder of the main chain, has control over the entire Bitcoin network. If an attacker is able to create such a chain and transfer the created chain, it will be accepted by the network as the main branch of the transaction database. The transactions contained in the previous main branch,

not in the one created by the attacker, are no longer confirmed by being added into a block by a miner and are therefore untrusted.

As a result of building a new main branch for the block chain, attacker is able to reverse the transactions which he signed and added into the previous main branch, back until the attacker split the chain[22].The attacker does this by simply not adding the transactions to a newly established branch and possibly using the same coins to issue other transactions, in which double-spending them as a result. For the network in which the chain with the highest computational cost associated is trusted, the older transaction of the same coins never existed and the recipient of the coins in the transaction who now never gets confirmations, loses his right to spend them as his transaction will never exists in chain again, since the coins are already spent and value is being signed to another recipient instead of him.

However, the attacker is unable to reverse transactions that are not sent by him because he does not know the private keys with which he signs the value over to other recipients. He also would not be able to create value out of nothing, he must follow proof-of-work and difficulty rules for building the blocks, even when generating an alternative attacking branch of the block-chain, otherwise it will not be accepted by other nodes. Attackers can not take money from other people because any transactions he add to blocks that have not been validly signed will not be accepted by other nodes in the network. In addition, these invalid transactions added to the block would also make the block unacceptable[4].

2.2.2 Denial of service attack

What an attacker will do is not take the transactions into his branch. Until added to the block chain these transactions would wait unconfirmed. This can be the case if the attacker loses the majority of the computing power on the network, stops his attack efforts, or includes others transactions into the transactional database created.Then, the transactions would receive the required level of confirmations to be trusted, and the transaction would be valid except if someone can fork the chain with his processing power and generate another branch after the previous split has become the main chain and before the transaction is added to a block in the main chain.

This can lead to a denial of service. Attackers can choose which transactions will be added into the chain. In fact, they can only add redemption transactions in their blocks that prevent all traffic passing on the value in the Bitcoin network making the system worthless. If users can not spend and receive coins, the currency is not attractive for them. In this way,

the attacker also loses transaction fees, but may not be worried, as the target in this case most likely to interrupt the growing popularity of Bitcoins. If they keep the control long enough, they can eventually stop using the currency completely.

The attacker who is in control also prevents other miners from mining valid blocks during the time they have the most computational power, since the other mining effort is put into branch which loses its status as the main branch in the block chain. An intelligent attacker would quietly build his chain in the background and not send the created blocks to the network. They need more computing power than the Bitcoin network combined to build this in the background. Once they unexpectedly broadcast their efforts to network, their chain is accepted by the Bitcoin protocol as the primary chain. If this attack is carried out an extended period of time, attackers may lose all processing power if honest nodes have passed it and they are unable to keep up. Then all of his efforts will be useless and it is very likely that the Bitcoin community will never know that an attack has been launched. In the same time, the longer the control period for the attacker, the greater the damage to Bitcoin. A few hours of unconfirmed transactions would not create chaos, but more than a week of reverse financial activity would cause average users to lose confidence in the system.

2.2.3 Difficulty of attacking with computing power

How difficult is it for an attacker to keep the computational effort high enough to generate an alternative chain that would be accepted as valid branch of the block chain? The current hashrate of Bitcoin that all miners produce together is 52,640,334 terahashes per second, which is equivalent to 6.6×10^{20} floating point operations per second (FLOPS) [7]. In the same time, the performance of the world's top 500 most powerful supercomputers together is about 1.4×10^{18} FLOPS[23]. This means that even if someone was able to mine Bitocin with these 500 supercomputers at full power, he only can discover an average of 0.0002% of the block and this would not be enough to control the network at all.

For an attack using computing power to succeed one of two conditions must be met at least: a highly motivated attacker with a tremendous amount of resources or decrease in mining activity. The motivated attacker need specifically target Bitcoin with highly probable destructive objectives, rather than keeping an eye on financial gain. Parties such as alternative currency systems or governments could, for various reasons, be behind such attacks, which are not discussed here.

The decrease in mining activity is quite possible in the future. Due to the amount of the reward for the discovery of a block half about every 4 years also decreases the mining incentive. In the same time, due to falling money supply Bitcoin is deflationary. This means that the motivation for theft and profitability of putting massive amounts of computing power for double-spending by generating alternative chain of transactions database rises[2]. So for attackers, they need less to attack and will get more profit if attack is successful. This condition could be right condition for attackers.

2.2.4 Mitigations for attacking with computation power

Although this attack on the history revision requires an incredible amount of processing power and there are not many parties that could afford such a mission, the threat is real and there are not enough mitigations. The only real defense against this attack today is that launching this attack is extremely difficult in terms of computing power required and with the same power honest mining would be more profitable for an attacker.

If somebody managed to launch an attack using computing power and could keep double the amount of hashing of the entire Bitcoin network together with next two years, he would theoretically be able to split alternate branch from the Genesis block and present it with a higher amount of the associated processing costs rherefore revise the entire transaction history [2]. In reality, because there are checkpoints in the current main chain that are hardcoded in the client software, rewriting the whole history would not work .

Hashes of trusted blocks are added with each new version of Bitcoin software [24]. Before another checkpoint is added to the software, attacker need to fork the chain from the last official checkpoint and merge it. Finding such an attack is relatively trivial: if a large number of blocks that were previously part of the main chain are now orphaned and multiple transactions that were previously confirmed are not confirmed now then most likely someone has successfully launched an attack.

To mitigate attacks using great amount of computing power, a combination of 2 kinds of methods can be work: protocol enforce fees or use of technical means rather than hard-coded checkpoints in trusted chain. The idea is that Bitcoin is protected by miners even if rewards of mining disappear, due to transaction fees they charge for transactions added to

the block they have hashed and broadcast to the network. For this to be true, Bitcoin popularity should continue to increase, and even then enforcement of fees could be introduced, as mining need to be worthwhile for participants to continually spend large hashing powers to make attacks more difficult. With voluntarily low fees, the transaction volume would have to be large, so that the miners can break even with the electricity costs which are used for constant Hashing. However, this mitigation could be easily introduced by the miners themselves: if some of the larger pools only begin to add transactions to blocks to confirm after a certain percentage of the fee is met, they enforce the rules to the network.

Bitcoin developers are very careful when adding checkpoints, they do not add them to newly discovered blocks which have not yet been proven to be part of the main chain. This is therefore not a real-time mitigation, although the possibility to rewrite the entire history of transactions with empty database is eliminated. Better technical defense mechanisms against attacks using computing power that allow constant control over branching without centralizing the currency are more difficult. In fact, the mitigation introduced by the developers is not really a decentralized way, as the community must trust the people who hard-code the checkpoints and therefore is not part of the pure Bitcoin protocol[24].

Additional decentralized peer-to-peer mitigation against attacks using computing power could be adding automatic checkpoints. If a client believes that the block is now trusted, it will be considered as a part of the main chain, even if there is a branch in later that exceeds the total processing power. The incoming packets are rated by clients based on a combination of metrics but not limited to them. The mining difficulty, the difference of the time the block was received and the discovery timestamps from reception. The miner who discovered the block, giving credit to entities which known to regularly add good blocks to main chain but without mining the previous or next block. Number of valid transactions hashed into block and rating of the blocks added to the same chain after that block.

If implement checkpoints in original Bitcoin software, it would use majority voting. If a block is rated by most nodes in the network have as high enough that it can be considered a checkpoint, an attacker can no longer fork a branch before that checkpoint, as most network members will not accept their blocks.

2.3 Cancer nodes

An attack on the Bitcoin network or target users with cancer nodes would mean that to fill the network with nodes controlled by the attacker. The goal would be to make a target of attack to connect only to malicious nodes or to separate part of the Bitcoin network from others. Flooding of the network with cancer nodes resulting that an attacker can reject to forward blocks and transactions which create denial-of-service attack. If attacker can segment the network as well, he can create the condition that several block-chain branches are built at the same time without being aware of the existence of others [22].

If an attacker successfully split the network by running a large number of cancer nodes, the attacker can double-spend coins easily similar to the method explained in attacks with computing power with very little effort. They would make a situation in which part of the network is built on top of 1 branch and trusting the transactions within that chain, which they believe are part of the main branch. In reality, after disconnected from the cancer nodes and the network finds there has been a folk in the block-chain and it will be resolved by choosing the branch with biggest amount of total computing power as specified by the protocol. Now the transactions in orphaned blocks remain unconfirmed, and the attacker may have been able to spend related coins in another branch.

If network segmentation is not completed, the attack with cancer nodes will fail. If the user whom attacker would like to disconnect from the network connects to an honest node, which is in turn connected to a network by at least one non-malicious node in it, he will receive enough information about transactions and blocks discovered to stay secure. This makes the total segmentation attack quite unlikely successful, as the disconnected parts of the network may not have only single connection.

There are already mitigations for cancer node attacks. In particular, Bitcoin clients make only one outgoing connection per 16-bit IP address network range [22]. This means that out of 65,536 addresses, for example, from x.y.0.0 to x.y.255.255, only 1 is used by the client to connect to the Bitcoin network. Therefore, an attacker would like to flood the network with cancer nodes would need to have control over several computers with IP addresses in a huge variety of different network range. This is possible if an attacker has access to a large botnet.

Another possible mitigation would be to use trusted auditing nodes with static IP addresses that clients can specifically connect to. These nodes can connect and keep block-

chain up-to-date. It could also detect if announced block-chain branches were created by attackers using huge amount of computing power. However, this trust network within the Bitcoin network violates the protocol and the idea of not trusting anyone in the peer-to-peer financial system. It is also possible that the honest trust nodes can be compromised and this can lead to chaos.

2.4 Client-side attacks

As we have seen, Bitcoin is very secure against attack as a system. For successful hackers it is still a profitable target as a financial system therefore the attacks are directed to the clients. Attacking the clients is achievable because in a decentralized currency like Bitcoin, users have more responsibility for gaining control of their finances. Securing the user's finances is left to the users themselves, since there is no centralized business to control Bitcoin.

There are several client-side attacks including wallet theft, denial-of-service, attacks on users anonymity and client software exploits. Here we define clients as both end users and also Bitcoin companies, such as currency exchange and discuss more common ways in which attackers can steal money or otherwise hinder the use of Bitcoin.

2.4.1 Wallet theft

As mentioned earlier, the Bitcoin wallet is a file stored on the user's hard drive. This file contains the keys required to receive and more importantly, for an attacker, send the bitcoins that are stored on the computer being accessed. Holding this file means that holding someones Bitcoin balance and owning control over their finances. This file can be accessed by breaking physical security or making contact with a device that has the wallet. In most cases, however, it's remote network activity and the use of malware to help criminals to hijack Bitcoins.

The first malware which targeting Bitcoin was Infostealer.Coinbit, a Trojan horse that lures users to execute it. When running, it searches Bitcoin Wallet in Windows computers and sends it by e-mails to the attacker via a server in Poland [25]. Symantec reported the attack during the Bitcoin bubble in June 2011 [26] and probably the 25,000 BTC

heist mentioned in the transaction chapter was executed using this malware. After trivial Infostealer.Coinbit, which targets Windows users, other malicious programs have been detected by anti-virus companies such as DevilRobber Trojan, which targets Mac computers and spreads pirated software downloaded from torrent sites. This malware is far more complicated. It steals wallet files, but also mines bitcoins, collects system information such as shell and browser history, collects usernames and passwords[27]. This means that with more complicated Bitcoin theft attempts, encrypting wallet files may not prevent the infected users from being robbed, as the malware can also set up a key logger and obtain encryption keys.

Users can now encrypt their private keys using the standard Bitcoin client version 0.4 or later. Shortly after the theft of 25,000 BTC this feature was added , and users can choose to use wallet encryption with the Advanced Encryption Standard symmetric key algorithm. When sending bitcoins users must enter their passphrase if keys are encrypted [28]. This reduces some of the simpler attacks because hackers must brute-force the encryption passwords to get to the private keys used to spend Bitcoins. However, this is not a big hurdle for a motivated attacker if user used weak passphrase. As mentioned earlier, some malware can also get the passphrase used for encryption, so encryption can provide some false sense of security to some extent and When users lose their secure passwords, they also lose their bitcoins.

In general, the use of Bitcoins in terms of client-side security is not significantly different from the use of bank or e-wallet systems for users: it is not safe to use unpatched, unsafe machines and compromised devices will result in loosing money. Users should not open suspicious files, browse shady websites, keep their software up to date, and be somewhat paranoid with the computer connected to Internet.

The Bitcoin protocol supports multiple signatures over transactions. This means that to authorize a transaction different private keys can be combined and before the requirements of the script section of this outputs are met, previous transactions outputs cannot be used for new transactions. Theoretically, it is even possible to use a combination of keys so that keys A or B and C are used to send coins that are sent to addresses which support multi-signature security or an even more difficult multi-key scheme [4]. This enhancement, for example, makes it possible to issue a transaction from a computer and then receive a notification on a smartphone to confirm the transaction, making the wallet much safer. Multi-level

authentication reduces the risk of being victimized by wallet theft, but makes use of Bitcoin more difficult and like wallet encryption, must be activated by users.

It's not only end users who have wallets which are good target for attackers. In addition to the world largest Bitcoin exchange Mt.Gox Hack, there were other high-profile attacks on Bitcoin services, some targeted specifically at wallet files. There was notable attack by hackers to Bitcoin business Bitcoinica, an exchange that allows forex-like market actions with contracts on rate difference and offers the opportunity to sell short the Bitcoins that is not owned by users, the deal is backed by their US dollar. Bitcoinica has lost its wallet files twice within three months. First, along with seven other Bitcoin wallets their wallet was stolen from Linux cloud provider Linode, whose customer support interface was exploited, and the stolen support credentials were used to compromise the accounts on Linode which is running Bitcoin clients were to serve their customers [29]. Bitcoinica was successfully attacked for the second time on its rackspace virtual server and lost the balance on its hot wallet, with which requested withdrawals were automatically paid. The service also lost the account information and transaction history for the attacker because they were deleted when the server instances were destroyed and no current backups were created [41].

Hot Wallet is the wallet that is stored on the online server and used for automatic transactions. This means that encryption and other simple mitigations to avoid money loss are in most cases not helpful, since attackers who have access to this wallet most likely compromised the server and could extract the encryption scheme from the source files or network traffic.

To avoid being hacked, Bitcoin service providers should protect both their public web applications and the servers and network. For servers, it makes sense to restrict both physical and virtual access to a minimum number of people, especially those outside the company. This means that the use of cloud services and virtual hosting providers should be avoided, as the temptation for employees of such companies to obtain a large amount of Bitcoins that are less likely to be punished might too big to resist. Bitcoin companies need to be aware that they are dealing with financial systems and therefore need to conduct continuous security audits and use third-party security audits to monitor their security.

There is one big non-technical security problem for Bitcoin services. To start developing Bitcoin companies is quite simple: there are many open source projects and code samples, as well as a helpful and intelligent community that you can get support from. In

addition, in most of countries no licenses, laws or regulations have to be followed to accept Bitcoins as a means of exchange or to offer financial services to Bitcoin. Low access barriers to start Bitcoin service providers can also result in poor software quality and security levels, as developers may not need a security background and there are no requirements to involve security-conscious individuals and conduct audits. In fact, there is no one in the peer-to-peer Monetary System who could enforce any rules.

Unfortunately, security breaches of Bitcoin companies and individuals who are victims of theft lead to bad publicity for Bitcoin as a system. In spite of the fact that the protocol itself is fairly secure, the public image is presented as a somewhat dangerous financial system. Experienced Bitcoin users can reduce the threat of falling victims of hackers and should carefully choose the services they trust with their money. The good thing about Bitcoin is that it is ultimately a peer-to-peer currency and user do not have to trust any of the service providers like a bank to take part in financial transactions. Users can simply run the Bitcoin client on their local machine, manage the transactional database, and validate all transactions themselves by running the software automatically.

2.4.2 Attacking anonymity

A big interest in Bitcoin comes from the perceived anonymity of Bitcoin transactions and the fact that you can send money online without restrictions, without revealing your real identity. This is very important feature for criminals like drug dealers, but also for people who are oppressed by their governments or just respect their privacy. For people who want to remain anonymous for some reason, they must understand that Bitcoin anonymity is pseudo-anonymous. The perception of anonymity is based on the fact that there are no registrations or credentials to join the Bitcoin network and issue transactions. Coins are related to addresses which look like random strings. In the same time, all transactions in the block-chain are publicly available and it is therefore possible to attack the anonymity of Bitcoin users. It might be law enforcement agencies to use it to find criminals who use the currency but it can be used by criminals as well to find and identify wealthy people who have huge amounts of Bitcoins.

To associate bitcoins with an identity, there must be an association point. One or more transactions or addresses must be able to be linked to real objects. This can happen when an IP is connected with a transaction, when goods are shipped through a shipping address, forum signatures are created with bitcoin addresses, registered to service sites and

assigned an address, or when money is sent from currency exchange sites which require personal documents or many other means. By combining this information it can create a mapping and add notes to the Bitcoin flow in transactions to identify real people using the coins.

By graphing the network and adding publicly available information with links from blockchain and open source intelligence, it is possible to link many public keys together and link the information with data outside of the Bitcoin network [31].The result of analysis, breaking anonymity is in practice a graph with points as addresses and links between them as transactions. The addresses themselves may be further investigated if they are in any way associated with persons or services through information already obtained. If a party with some authority were to carry out such mapping, they could probably retrieve the data for user information from currency exchange sites as well as from other services and thus get a more complete picture and possibly even name thieves if the hackers did not do it cautious enough To take steps to stay anonymous.

Even though Bitcoin is implemented as not very anonymously, it is possible to stay one step ahead of attacks on anonymity. Bitcoin users can use as many number of addresses as they want. User can uses a different address for all transactions, so the mapping completion fails for an attacker, and the gaps in his information diagram can hinder the process of unmasking a Bitcoin user. Users who have a strong need for anonymity also want to use Tor or similar services to hide network traffic and the location of their computers. Persons wishing to remain anonymous should be particularly careful when sharing information with services related to Bitcoin. A real possibility to assign addresses to identities emerges if the linking-point has been created, For added anonymity, there are Bitcoin mixing services: they take the coins sent by users and mix them with the coins of other users and then return them to another address of user, try to confuse the Bitcoin flow trail and increasing personal anonymity[31].

In addition, Bitcoin traffic is not encrypted [2]. The data send via the peer-to-peer network is plaintext, even though the system itself uses strong cryptography. It does not create opportunities for man-in-the-middle attacks, as the falsification of ECDSA digital signatures is currently infeasible, but raises some additional security concerns. This may particularly affect the anonymity of the users.

There is constant Bitcoin traffic to and from the computer running the Bitcoin client since Bitcoin users receive and forward new transactions they receive from the network. The first person who announce a transaction is the person sending the coins in that transaction. Other nodes trap the packet with the transaction and then forward it to their associated nodes. Some of these nodes mining coins and they add the transaction hash to the Merkle tree. It will be included into a block if lucky enough, which will then be announced to the network. It then creates new blocks on top of it. that confirmed and enhances confidence that this transaction can not be undone by an attacker with high computing poewer.

The first person to send information about a transaction therefore also specifies their bitcoin addresses. This can be a great mapping point to associate real identities with bitcoin traffic and addresses. They must have three pieces of information to be able to make this mapping : a good overview of the network, and specific Bitcoin traffic, the traffic of the nodes connected to the client, and the personal information of the person they want to investigate. While anonymity attack using this method can be performed by using cancer nodes to get a good picture of the transaction flow in the network connected to a specific client, there is a better chance of reducing anonymity by tapping the network to monitor all traffic passing one node or better yet several related nodes. Someone who is capable of performing such an attack is likely to want to work with an Internet service provider who also knows the name and actual location of the network owner.

To map a Bitcoin user to real identity turns out to be difficult if he's really worried about his anonymity, but with enough motivation, resources, and connections, this is possible. However, this is not a problem for average Bitcoin user and the goal of Bitcoin design is not to be really anonymous.

2.4.3 Denial of service attack and client software security

Several methods can be used to derive a denial-of-service against one or more target users, but in some cases for the entire Bitcoin network. As explained in attack with computing power and cancer nodes, these are the theoretical but somewhat impractical. To target a user to disconnect from the Bitcoin network, this could also result in exploitation of a vulnerability in Bitcoin client software. By finding defects in open source software, an

attacker may overflow the client to shut it down or even worse, send data to it which enable code execution which could fetch private keys if it's not encrypted.

Denial-of-service attacks to disable client software would mean that to send the node running the client a large amount of information or specially crafted inputs which are not processed properly. Attackers send too much data too quickly or invalid transactional messages which will cause targeted node to lose their connection. To prevent it, Bitcoin client has integrated denial-of-service prevention [4]. This defense can be bypassed by quickly sending data from multiple malicious nodes, however, as mentioned in cancer node attacks there is the restrictions on connections per IP space to prevent it.

To disconnect a node from the Bitcoin network, finding a vulnerability in the client software could be a better chance for an attacker. No software with some level of complexity is completely safe against attacks. For Bitcoin being an open source project which increases security by two different views. First of all, anyone can read the code and look for malicious cases that have not been properly handled or find other types of vulnerabilities. In the same time, users reviewing the code can also report and resolve the issues that have been identified.

A critical vulnerability was reported in Bitcoin software in May 2012 and identified as CVE-2012-2459. This vulnerability allows attackers to isolate target from the Bitcoin network and result in creation block-chain forks[32]. The denial of service would have affected almost all users running the default client. It was reported and silently fixed by patching major Bitcoin mining pools and services software before the discovered flaw and security hotfix were made public.

A good case of responsible disclosure, as well as quick attention and resolution of issues, demonstrate the maturity of the Bitcoin project and the skills of the core developers involved. However, this does not mean that client software is safe and invulnerable always. As the software itself becomes more complex to support cases such as multi-signature transactions or other new features, the attack surface also increases. In the same time, attention is growing on the system and motivation for attackers due to the increasing possibility of financial profit on successful exploits.

If an attacker is able to take the nodes offline, the Bitcoin software can be restarted and connected to the network. After the release of patches for such vulnerabilities, the network continues to operate only with a small financial loss. The loss would be for miners

who lose the opportunity to seek for block solutions. With less mining competition and less difficulty, the attacker would be able to mine more blocks and receive the mining fees while several miners are constantly offline. It would also make it easier for the attacker to run attacks using computing power and double-spend. To launch such a large denial-of-service attack, an attacker would need to find 0-day vulnerabilities in Bitcoin client and supporting infrastructure to constantly send exploits to nodes in the peer-to-peer network. Part of the mitigation against this attack is provided by using several Bitcoin client software that connect to the Bitcoin network, as it is very unlikely to find exploits for all available clients.

As the damage caused by denial-of-service conditions in client software is rather small and temporary, it's not the ones we should be most concerned about. However, there is always the possibility that an attacker might one day find a way to run code remotely by using a software vulnerability. To install malware, send bitcoins, or steal keys, a buffer overflow or similar anomaly can be maliciously exploited. This would not be a defect in the Bitcoin protocol or the system design, but could have serious consequences for the entire network. If a method to create a transactional message which would trigger something unexpected in the client software found by an attacker, it would spread across the network and potentially affect all Bitcoin users.

Of course, the solution to this is similar to approach of other open source software projects: Experienced users should proof-read the source code, write good test methods to identify corner cases where unexpected input causes problems and implements solutions that may cause problems. There is no security guarantee for open source software, and for Bitcoin, a exploitable vulnerability has a greater impact than many other systems because of the potential for rapid propagations in the peer-to-peer network and the financial importance of the system.

2.5 Summary of attacks

Hash collisions are most likely not going to occur and the client software include mitigation against related threats. Creating key pairs where the resulting addresses collide is impossible unless quantum computing or cryptanalysis make a major breakthrough. Theoretical attack scenario, such as using vulnerabilities in ECDSA to find out a private key

comes from public key fails because getting the public key requires preimage attacks on two different hash functions which currently considered as unbreakable. Preimage attacks or hashing algorithm enhancements have an impact on mining, allowing attackers to start an attack using computing power with less resource use. However, these methods could not be used for stealing coins or shutting down the system on their own, and weakening the hash functions is not a threat to the integrity of block chains. Cryptography used in Bitcoin is very strong and its significant weakening is unlikely in the near future.

Because of its high-security system overall Bitcoin, sees many attacks targeting both client systems and end-users and similar businesses. Attacking the anonymity of users by analyzing and mapping the public transactional database to external information is possible, but it's very difficult if the user acts carefully. However, this is not a problem for Bitcoin security because anonymity itself is not a design goal. As explained in wallet theft, we concluded that trading in Bitcoin clients need to be particularly careful about the security of their systems, that they have limited trust in as few parties as possible, and that multiple signature transactions will provide the client-side with significant added security. We are not safe from security vulnerabilities discovered in the Bitcoin software, but the overall security of the software is good and the community has shown that it is able to handle potential problems.

The attack with a huge computing power is still the main concern of Bitcoin as a system. It's very hard to start, but doable and the problem is that there are no mitigations against it. For a mitigation, rating blocks and the automatic addition of checkpoints to the block chain have been proposed. This mitigation would also significantly reduce possible damage by cancer nodes and reduce the threat of large-scale denial-of-service attacks that stop Bitcoin system. Another suggestion was to add features that could force the Bitcoin software to connect to trust nodes in the network to reduce the risk of getting trapped between cancer nodes and being denied sending or receiving valid Bitcoin transactions or finding reverse transaction.

3 USECASE OF BITCOIN

Bitcoin bubble in 2017 bring Bitcoin big interest from people who was not interested in cryptocurrency because of its cryptographic and peer-to-peer features which make non-technical people difficult to understand and trust it. Bitcoin bubble was disappear in few weeks but more and more people started using Bitcoin after that. However besides its popularity, using Bitcoin to consume goods is not very popular way of using it yet. There are several coffee shops and Hotels accepting Bitcoin, but they are mostly using it to promote their business. Since there is still not a lot companies accepting Bitcoin in 2019, it will attract people attention and to prepare to accept Bitcoin is very easy and cheap or even free. Currently the most popular use case of Bitcoin is for investment. There are large number of Bitcoin exchanges everywhere in the world because of its popularity and also easiness to participate since there is several countries have no restriction to open Bitcoin exchange. However since Mt.Gox hack, a lot countries governments started realize that there was massive amount of money was invested to Bitcoins and its effect to people and started to regulate to open Bitcoin exchange nowadays. Therefore some of those Bitcoin exchanges, however not all of them, like Coinbase, Circle, Gemini and Xapo are insured by the FDIC, up to 250000 US Dollars.

Another popular way of using Bitcoin is to transfer money to foreign country. Bitcoins features, low fee and quick processing time are very big advantage when user transfer money to abroad since other foreign money transfer services right now is expensive and takes time for money to arrive from sender to recipient. I myself have used Bitcoin to transfer money from Taiwan to Europe. First sender, my friend, buy a Bitcoin from local person using Finnish Bitcoin exchange service LocalBitcoins.com, and transfer that coin to recipient, me, then I sell it on LocalBitcoins.com.

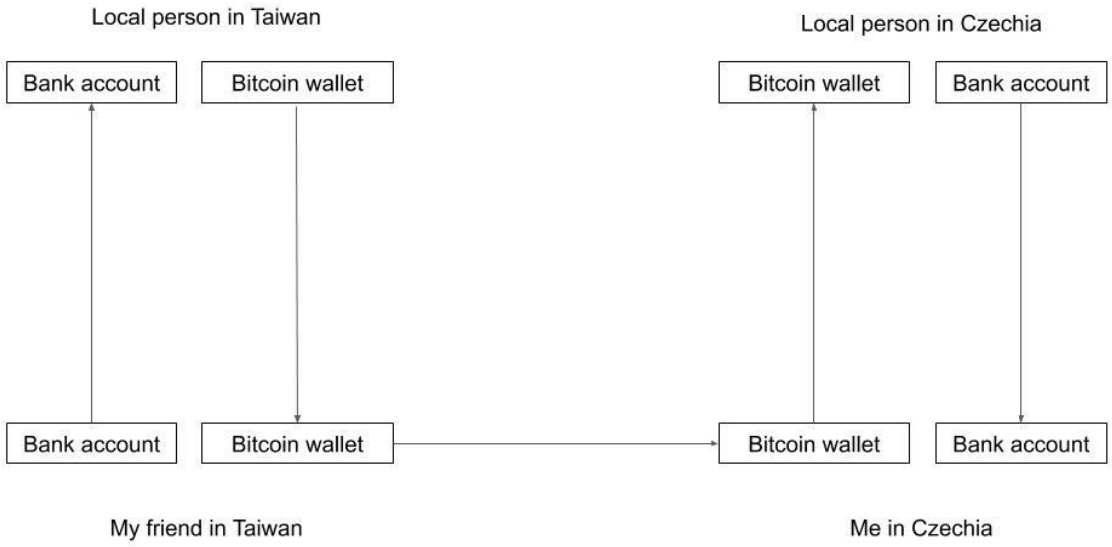

Figure 11: Transfer money from Taiwan to Czech Republic

The cost for sending money is only bank transfer fee when my friend bought Bitcoin from local person which can considered as nothing compares to traditional services to transfer money to abroad like Transferwise, which actually takes very low fee compares to other services. And execution time for our transfer was about 24 hours, since we exchange Bitcoin to traditional currency but if it was only transfer of Bitcoin, it could take less than one hour. As online shopping industry grows very big and buying goods from abroad becoming more and more popular along with it, using Bitcoin to send money to sellers in foreign countries could become very popular as its low fee and simplicity to participate. It could be the most popular way for normal people to pay to online shopping service.

CONCLUSION

In this paper We have described principle of Bitcoin, its security and usecase. We have shown that even though cryptography used in Bitcoin is not breakable but the system can be attacked with a huge computing power or cancer nodes. However, these attacks are very difficult and in fact hackers are targeting Bitcoin clients to steal their wallets by using software vulnerability. We also showed that Bitcoin is not 100% anonymous, but a user who wants to keep their identity secret has the opportunity to increase their chances to keep their identity secret.

For Bitcoin extensions and additional security, we offer ideas for monitoring nodes on the network that prevent clients from trusting the attacker-generated Branch of Transactions database. We provide a way to mitigate potential problems with attacks with a huge computing power by rating blocks and adding checkpoints to the block chain. Future work should include further investigation into the mitigations and their implementation, as well as mathematical evidence of Bitcoin cryptographic security.

For businesses or individuals who want to trade bitcoins due to low fees and restrictions, high control over their finances and convenience, system security is very good. From a technical point of view, it can be used safely to send, receive and hold large amounts of value. However, all users must be aware that with increasing power over their finances, they also need to take greater responsibilities and take steps to protect both personal and corporate systems when trading Bitcoins. For larger business cases, it is strongly recommended that you work with security-conscious people and conduct regular security reviews of your services.

BIBLIOGRAPHY

[1] Bitcoin (Oct 23, 2018), from *https://www.investopedia.com/terms/b/bitcoin.asp*

[2] Andreas M. Antonopoulos (20 Dec. 2014) Mastering Bitcoin

[3] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder (Feb 9, 2016), Bitcoin and Cryptocurrency Technologies

[4] Aaron Cunningham (July 3, 2018), Checksum Addresses: How Blockchains Add a Layer of Security from *https://news.coinsquare.com/learn-coinsquare/checksum-addresses-how-blockchains-add-security/*

[5] Jordan Baczuk (Jun 6, 2018) How to Generate a Bitcoin Address—Step by Step from *https://medium.com/coinmonks/how-to-generate-a-bitcoin-address-step-by-step-9d7fcbf1ad0b*

[6] Jon Martindale (Sep 18, 2018) What is Bitcoin mining? from *https://www.digitaltrends.com/computing/what-is-bitcoin-mining/*

[7] Ittay Eyal, Emin Gün Sirer (July 7, 2018) Majority Is Not Enough: Bitcoin Mining Is Vulnerable from *https://cacm.acm.org/magazines/2018/7/229033-majority-is-not-enough/fulltext*

[8] LUKE FORTNEY (Feb 10, 2019) Bitcoin Mining, Explained from *https://www.investopedia.com/terms/b/bitcoin-mining.asp*

[9] Eduardo Cruz (Oct 2, 2017) What is a Satoshi? Bitcoin and its 8 decimal places. from *https://medium.com/airtm/what-is-a-satoshi-bitcoin-and-its-8-decimal-places-cffeb5795758*

[10] Daniel Jeffries(Jun 23, 2017) Why Everyone Missed the Most Important Invention in the Last 500 Years from *https://hackernoon.com/why-everyone-missed-the-most-important-invention-in-the-last-500-years-c90b0151c169*

[11] Dan Kaminsky (Apr. 12, 2013), I Tried Hacking Bitcoin And I Failed from *https://www.businessinsider.com/dan-kaminsky-highlights-flaws-bitcoin-2013-4*

[12] Lane Wagner (Jun 29, 2018), (Very) Basic Intro to Hash Functions (SHA-256, MD-5, etc) from *https://blog.goodaudience.com/very-basic-intro-to-hash-functions-sha-256-md-5-etc-ed721622ff8*

[13] Erick Stingaciu (Mar 14, 2019) What is a Block Header in Bitcoin? from https://coindoo.com/what-is-a-block-header-in-bitcoin/

[14] Christoph Dobraunig, Maria Eichlseder, and Florian Mendel (December 03, 2015) Analysis of SHA-512/224 and SHA-512/256 from *https://eprint.iacr.org/2016/374.pdf*

[15] Xiaoyun WangYiqun Lisa YinHongbo Yu (August 18, 2005) Finding collisions in the full SHA-1 from *https://www.iacr.org/archive/crypto2005/36210017/36210017.pdf*

[16] Florian Mendel, Norbert Pramstaller,Christian Rechberger, Vincent Rijmen (September 02, 2006) On the Collision Resistance of RIPEMD-160 *from https://www.esat.kuleuven.be/cosic/publications/article-1355.pdf*

[17] Graham Templeton (July 29, 2015) What is Moore's Law? From *https://www.extremetech.com/extreme/210872-extremetech-explains-what-is-moores-law*

[18] Daniel R. L. Brown (January 27, 2010) SEC 2: Recommended Elliptic Curve Domain Parameters from *https://www.secg.org/sec2-v2.pdf*

[19] Yang, E. Z (May 13, 2012) The Cryptography of Bitcoin from http://blog.ezyang.com/2011/06/the-cryptography-of-bitcoin/

[20] Christoph Dobraunig, Maria Eichlseder, and Florian Mendel (December 03, 2015) Analysis of SHA-512/224 and SHA-512/256 from *https://eprint.iacr.org/2016/374.pdf*

[21] Arjun Kharpal (JUN 18, 2018) ng you need to know about the blockchain from *https://www.cnbc.com/2018/06/18/blockchain-what-is-it-and-how-does-it-work.html*

[22] Benjamin Fabian (May 2018) Adoption of Security and Privacy Measures in Bitcoin - Stated and Actual Behavior from *https://www.researchgate.net/profile/Benjamin_Fabian/publication/325300680_Adoption_of_Security_and_Privacy_Measures_in_Bitcoin_-_Stated_and_Actual_Behavior/links/5b045f1e4585154aeb07f280/Adoption-of-Security-and-Privacy-Measures-in-Bitcoin-Stated-and-Actual-Behavior.pdf*

[23] PERFORMANCE DEVELOPMENT (May, 2019) from *https://www.top500.org/statistics/perfdevel/*

[24] Thin Client Security (22 May, 2018) from *https://en.bitcoin.it/wiki/Thin_Client_Security*

[25] Dan Goodin (18 Jun, 2011) New malware ferrets out and steals Bitcoins from *https://www.theregister.co.uk/2011/06/18/bitcoin_stealing_malware/*

[26] Symantec (June 16, 2011) Infostealer.Coinbit from *https://www.symantec.com/security-center/writeup/2011-061615-3651-99*

[27] Peter James (Oct 28th, 2011) New Malware DevilRobber Grabs Files and Bitcoins, Performs Bitcoin Mining, and More from *https://www.intego.com/mac-security-blog/new-malware-devilrobber-grabs-files-and-bitcoins-performs-bitcoin-mining-and-more/*

[28] BitcoinCore (23 Sep, 2011) Bitcoin version 0.4.0 released from *https://bitcoin.org/en/release/v0.4.0*

[29] Steve Walters(Feb 15, 2019) Biggest Bitcoin Hacks: 7 of The Largest Breaches in History from *https://www.coinbureau.com/analysis/biggest-bitcoin-hacks/*

[30] Vitalik Buterin (Jul 17, 2012) Bitcoinica Stolen From… Again from *https://bitcoinmagazine.com/articles/bitcoinica-stolen-from-again/*

[31] Micha Ober, Stefan Katzenbeisser, Kay Hamacher (7 May 2013) Structure and Anonymity of the Bitcoin Transaction Graph from *https://www.researchgate.net/publication/272646209_Structure_and_Anonymity_of_the_Bitcoin_Transaction_Graph/download*

[32] NATIONAL VULNERABILITY DATABASE (08 Jun,2012) CVE-2012-2459 Detail from *https://nvd.nist.gov/vuln/detail/CVE-2012-2459*

LIST OF ABBREVIATIONS

BTC	Bitcoin
SHA-256	Secure Hash Algorithm 256
RIPEMD-160	RIPE Message Digest
ECDSA	Elliptic Curve Digital Signature Algorithm
CPU	Central Processing Unit
RSA	Rivest–Shamir–Adleman
SHA-1	Secure Hash Algorithm 1
ANSI	American National Standards Institute
FLOPS	FLoating point Operations Per Second
IP	Internet Protocol

LIST OF FIGURES

LIST OF TABLES

APPENDICES

APPENDIX A I: CALCULATION OF TIME TO BREAK RIPEMD-160 WITH CONSIDERATION OF MOORES LAW

This python code calculate the time needed to break RIPEMD with consideration of computing power improvement according to Moores law.

```
seconds_month = 3600 * 24 * 30      # seconds in month
hasherate = 40 * (10 ** 12)         # initial computing speed # hashrate per second
counter = 0                         # counter of month
hashes = 2 ** 80                    # average hashing attempts needed per second

# loop until hashes run out
while hashes > 0:
    hashes -= (hasherate * seconds_month)    # decreases hashes/month from total hashes
    counter += 1    # increase counter of month

    # update hasherate every 18 month
    if counter % 18 == 0:
        hasherate *= 2

print( counter / 12 )    # print out total amount of years take
```