TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Computer Science
Chair of Network Software

Robert Pallas

Bitcoin Security

Master's Thesis

Supervisor:
Rain Ottis, PhD

TALLINN 2012

Author declaration

I hereby declare that this master's thesis is written independently and has not been submitted previously. All external materials like literature and web pages used in the writing process are referenced within the document.

01.06.2012

Abstract

Bitcoin Security master's thesis gives an overview of peer-to-peer cryptographic currencies security model. This paper introduces Bitcoin and its design principles and cryptographic primitives used to create the system. Main focus of the thesis is on Bitcoin security. It seeks to develop a better understanding of how Bitcoin is secured by analyzing different attack methods towards the system, their relations and defence mechanisms in Bitcoin design. We show that Bitcoin uses strong cryptography that is currently unbreakable, but the system can be attacked with a lot of computing power and special attacking machines connecting to network. Those attacks are however a lot harder than client-side attacks due to users big responsibility levels in peer-to-peer currency. Client-side attacks include wallet theft, breaking the users anonymity and denial-of-service.

Annotatsioon

Bitcoini Turvalisus on magistritöö, mis annab ülevaate partnervõrgus toimiva krüptograafilise valuutasüsteemi turvalisuse mudelist. See kirjutis tutvustab Bitcoini disainipõhimõtteid ja süsteemi loomiseks kasutatud krüptograafilisi lahendusi. Peamine fookus on Bitcoini turvalisusel. Töö eesmärgiks on analüüsida Bitcoini turvalisust näidates erinevaid rünnakuid, nende seoseid ja süsteemi disainis implementeeritud kaitsemehhanisme. Näitame, et Bitcoin kasutab tänaste vahenditega mitte murtavat krüptograafiat, kuid seda on võimalik rünnata suure hulga arvutusvõimega ning spetsiaalselt ründamiseks võrku ühenduvate masinatega. Need rünnakud on aga kordades keerukamad kui ründed klientide vastu, sest partnervõrgus toimiva valuuta puhul on kasutajate vastutus suur. Kasutajate vastu suunatud ründed sisaldavad rahakoti vargust, anonüümsuse purustamist ja finantstegevuse takistamist.

Table of Contents

Illustration Index

1. Introduction

1.1. Problem statement

World has seen several fiscal crises during the last few years with a lot of governments having difficulties keeping their economies running efficiently and as a result there have been financial crashes that have affected the lives of millions of people. This has created a demand for new kind of niche money, digital currency that is not controlled by governments.

World has also seen a rapid development in information technologies. Internet has formed into a truly global system that most modern people in developed countries regularly use. The amount of services grows rapidly and the knowledge for developing complex systems is spreading. This has created a possibility for a new kind of decentralized digital cash system to emerge. It is called Bitcoin.

Bitcoin is the world's first decentralized digital currency[1]. Bitcoin users do not have to rely on banks or other central authorities to send and receive money via Internet, it works without middlemen. Therefore a user of this system must not trust other parties to manage their financial records. It uses peer-to-peer networking, digital signatures and clever cryptography to enable irreversible and fast international payments with low fees.

As Bitcoin has no central authority to trust its security lies solely on the design of the system itself. All systems have their flaws however and they are targeted when money is moving within the system. As a digital currency Bitcoin will take a lot of heat from both technical and non-technical attacks. Users who wish to adopt the system have to know how secure is trading in Bitcoins, what are the threats to the system and how should they act to keep their money as safe as possible.

This thesis looks upon the security of Bitcoin, identifying its main weaknesses and looking into ways how threats are or could be mitigated. Purpose of this work is analyzing different technical attacks against Bitcoin as a system, their relations and related defence mechanisms in Bitcoin design. Main attacks against Bitcoin as a system are outlined and analyzed in simple terms to be understandable to most people interested in Bitcoin security model with no or little extra effort outside understanding this paper. At the same time biggest security concerns in Bitcoin are discussed in enough detail to get an overview of the attacks, their difficulties and mitigations in place from various angles.

1.2. Outline of the thesis

The thesis is divided into 2 main parts: introduction of Bitcoin and Bitcoin security. In introductory chapter 2 we look at how Bitcoin system works. As Bitcoin is rather complex system this part describes it from a view-point that gives the readers needed background to understand the attacks presented in chapter 3. This means that we are mainly studying technical principles like peer-to-peer design, publicly held hash-linked chain that acts as a database of all transactions and use of public-key cryptography for sending transactions as those are the main innovations in Bitcoin system and also the most important properties for attackers. Introduction also includes Bitcoins history with short overview of previous works that have influenced creation of the currency and reasons for inventing digital cash systems such as Bitcoin. We end our introduction with looking at the most important cryptographic principles used for creating Bitcoin.

Chapter 3 focuses on possible weaknesses of Bitcoin. We give an overview on client-side attacks that target Bitcoin users like tackling their anonymity and stealing the wallet. Before that we present our main focus: technical vulnerabilities of the system as a whole. We look at breaking the cryptography, denial-of-service and double-spending attacks with help of lot of computing power and cancer nodes in more detail. Chapter 3 builds on top of knowledge gained from introductory part and within lies the main contribution: analyzing different attack scenarios and their relations.

Chapter 4 summarizes Bitcoin security and gives recommendations to users, Bitcoin businesses and developers.

1.3. Related work

Bitcoin is a young project and there is not much published research about the currencies working principles and even less so about its security. Most authors do not go beyond stating Bitcoin is secure as a system and to our knowledge there is no material that combines possible attacks on Bitcoin with researching their relations to each other and how they are dealt with using cryptographic principles.

A lot of the information about the design principles come from original Bitcoin white paper, knowledge about the protocol itself, used cryptographic primitives and their relations have to be read from the source code of the original client. For this Bitcoin wiki is an invaluable proxy. In peer-to-peer currency also the documentation is created collaboratively and wiki page is chosen by Bitcoin developers and enthusiasts alike to spread their message, introduce features in original client software and bring light upon inner workings of the system. For those reasons and lack of more authoritative and exhaustive materials on Bitcoin design wiki pages are extensively used to write Bitcoin papers.

2. Bitcoin

Bitcoin is a decentralized digital currency. It offers fast, secure and irreversible international transactions with low fees. Bitcoin transactions go from person to person not through a banking system. One user sending funds to another adds a digital signature to a transaction message saying that another person is now the holder of those coins and broadcasts it to network. Receivers of this message then send the message to other machines connected to them propagating the information over the Internet.

All Bitcoin transactions are known to users, they hold the complete transactions database on their computer and can verify financial dealings by themselves and do not have to trust anyone for the informations integrity and authenticity. Extra Bitcoins are created by solving computationally difficult puzzles. This process is called mining. Mining is defined by algorithms and creates a pre-determined and transparent money supply: it is always known how many Bitcoins are and ever will be in circulation. First one to solve a mining puzzle receives a reward, becoming the owner of newly created coins. Mining is also used for adding transactions to the database and avoid double-spending.

2.1. Peer-to-peer currency

Bitcoin differentiates from most of other virtual currencies due to its peer-to-peer design. There is no central clearing house or monetary authority run by a company or organization. It is also not fixed to any real money, although it can be used to purchase many real world products not just virtual goods and services. Instead of relying on central bank and giving it powers to monitor, control and approve transactions as well as manage the money supply, Bitcoin is underwritten by a peer-to-peer network similar to file-sharing services like BitTorrent[2]. All devices running Bitcoin software can therefore act as both client and server: sending and receiving information about transactions. Participants give away some of their bandwidth and disk space but overall the setup and running costs are very small.

Bitcoin wallet owners can verify all the transactions ever carried out making the system very transparent and easily auditable. The Bitcoin market has very low transaction fees which are optional and chosen by the payer[3]. It is however customary to include a small fee in every transaction and original Bitcoin client adds this automatically. Transfers that carry a fee are more likely to be added to the block-chain that is used to verify the transactions. Moreover, some of the Bitcoin transactions can carry low monetary value but a lot of data to process. To make confirmation of such financial dealings worthwhile, adding a fee is more than encouraged by Bitcoin community. Before transactions are confirmed its various components like digital signature of the payer are validated. After validation the coins used in the operation are checked against double spending and then the transaction is added into official records called blocks that are constantly added on top of each other in decentralized way[3].

The transaction messages are broadcast across the network and nodes can leave and rejoin the network any time, accepting the longest valid chain of blocks that contain transactions as proof of what happened while they were not connected[4]. Because of completely distributed architecture we have to assume that most of nodes in the network are honest. Majority vote mechanism is used for double spending avoidance and resolving any collisions of interest[3]. At first glance the need for assumptions is the biggest security issue Bitcoin has and there are interesting solutions for users to be able to trust the system like paying to people for being honest and adding transactions to blocks, verifying them and auditing the system.

Low fees are attractive in case of micro-payments where fees would dominate in case of centralized systems. Bitcoin is also appealing for sending and receiving money internationally since there is no intermediation entity who wants extra money for running their services and therefore no additional costs[3]. Decentralized design for a digital currency system lets us have means of trade where we do not have to trust a government. It gives us opportunity to privatize money and we can use it without going through currency exchanges, intermediary payment providers or other third parties that make the whole process of exchanging values too difficult, slow and expensive. Furthermore the money we use today is not designed with Internet in mind. We do have ways to deal with current monetary systems and global networks, but they lack in simplicity and overall security.

2.2. Bitcoin wallet

Bitcoin wallet is a file in users filesystem. It holds public and corresponding private key pairs and transactions done from and to this wallet. The keys are used to receive and send Bitcoins. Public keys are given to payers to identify receiving parties and private keys are used to sign transaction messages and confirm the exchange of currency. User preferences are also kept in those wallet files that can and should be encrypted to mitigate the risk of loosing the coins to a hacker.

A Bitcoin address is a 25-34 character identifier that consist of numbers and both upper and lower case characters. Most addresses in use are 33 or 34 characters long. The address usually starts with 1 and never contains either number 0 or upper-case letter “O” nor lower-case “l” or upper case “I” for better readability. The address itself is hashed from the public part of Elliptic Curve Digital Signature Algorithms (ECDSA) key pair which we will detail later. After several rounds of hashing with RIPEMD-160 and SHA-256 hashing algorithms a checksum for address is added and they are encoded with modified Base 58 encoding that results in mentioned format[5].

An example Bitcoin address would be 1N3rjCLXhuuWFCweLV88GrDym4pryx7tkq. This can easily be validated and sending Bitcoins to address that can not have corresponding private key and thus can not be used to receive the sent amount is pretty unlikely. The probability that a mistyped address is accepted as being valid is approximately 1 in 4,29 billion[6]. Therefore there is good protection for typing errors although typing the addresses in for sending Bitcoins is probably a rare occasion.

ECDSA key pairs and Bitcoin addresses are not part of Bitcoin network structure. They can be securely created offline following the hashing and encoding rules described in original design. The Bitcoin network will know about the address only after it is first used and has a transaction pointed towards it. As creating the addresses is easy and fast with several tools including the original Bitcoin client it is trivial for people to use several or even thousands of addresses to enhance their anonymity. Although sending Bitcoins to invalid addresses is impossible a transaction can be made to an address where wallet file with keys is lost due to owners careless behavior, machine failures or malicious activities. There is no feasible way to ever use those coins again and they are lost forever. This is another reason why it is wise for Bitcoin users to keep secure backups of their wallets.

Original Bitcoin client is written in C++ and is open-source[7], but several other pieces of software are available to connect to Bitcoin network and participate like Java client BitCoinJ where only headers of blocks are downloaded[8]. By design it is possible not to keep records of Bitcoin transactions for which all money received is already spent and therefore included in other transactions. Those measures are needed for Bitcoin to be scalable and to be used with high transaction volumes, ones comparable to credit card transfer rates in the world today. One may also choose to use eWallet services to avoid downloading the ever increasing block-chain with all transactions, but this results in giving away some of the control over their Bitcoins as well as possibly accepting higher fees on transactions.

2.3. Mining

Adding transaction records into journal of all transactions is called mining to refer to looking for gold, but more accurate term would be auditing as those who contribute computing power to find new blocks also check the transactions and help secure the network and the ones who contribute more power have better odds to receive new coins as a reward[9].

A block consists of one minting transaction that gives reward to the creator, zero or more regular spending transactions, a computational proof-of-work, a timestamp and a hash that references the chronologically prior block. Proof-of-work is a piece of data that is difficult to produce. It has costly and time-consuming working procedure that results in situation that satisfies certain requirements[10].

In Bitcoin world proof-of-work is a nonce, a value that is added to the block so that all valid transactions that have been announced in the network but not added to any of the previous blocks and reference to chronologically previous block can be hashed together to meet difficulty of mining any given time. The mining difficulty is constantly adjusted according to the time it takes to find the nonces and create new blocks. Goal of changing it is averaging 1 block creation every 10 minutes across the entire network[3]. Difficulty helps describe the target value of which the blocks resulting hashes integer value has to be smaller. Thats why hashes of blocks start with several consecutive zeros, bigger mining difficulty results in more zeros in the beginning of blocks hash.

Finding a nonce that would produce the hash of all the data in the block to satisfy the target requirement is computationally expensive process as it is only possible to solve it by trying out different nonce values. Proof-of-work and mining concepts eliminate the possibility of creating fake coins. Those would not be accepted by the network and it is therefore not possible to use them in transactions.

Mining for Bitcoins is a process with a lot of luck involved as the discovery of a block is a random event. As there are thousands of miners looking for free money the probability of solo miners running their special mining equipments finding a block and redeeming a reward is very small. The idea of free money is flawed of course since the equipment used for continuous hashing needs electricity and input power does not come without charge in most cases.

In total the network hash rate in May 2012 is about 12 terahashes per second[11]. This is also in a way network security rate as this is the amount of resources over which it is possible to think about attacking the network to reverse your transactions. As a single miner with standard equipment will on average generate 500 megahashes every second and have a probability of 70% of not finding a block within one year[12], the miners group together to share resources and the profit made. The concept of organized mining is called pooled mining. The pool operators get a fee for running the operation and participants divide the reward when one of them discovers a block according to amount of work someone has put into it in terms of hashes computed. This results in small and steady income instead of high probability of possibly getting no income at all in the competitive mining environment.

The blocks are linked together into block-chain. Every block contains a hash of previous one. Block-chain acts as transactions database and all nodes keep it up to date in their own machines by sending and receiving data of transactions and created blocks. For an example of this linear chain refer to figure 1, that shows the linking of blocks with references of previous blocks hash that is found as a result of mining by increasing the nonce value.

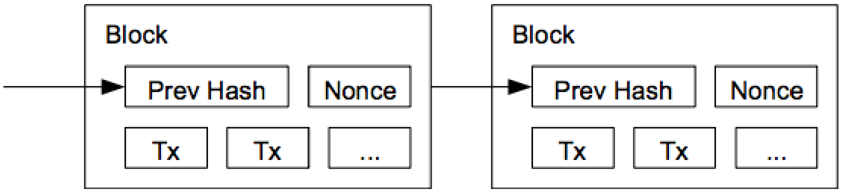

Figure 1: Block-chain [4]

Block-chain started with genesis block that is hard-coded in the software. The regular creation of new blocks has two reasons: it ensures that new transactions are added into collectively maintained database as fast as possible and is responsible for the creation of new coins. This means that mining is a decentralized process that helps keeping the Bitcoin system running while offering economic incentives to people giving away some of their computing power[3]. There are two types of rewards the miner who finds the solution to new block and adds it to block-chain receives: they collect the transaction fees and the reward for discovery which depends on the amount of blocks in the chain and is decreasing over time. Miners add the reward into newly discovered block as the first transaction where they claim ownership of minted coins.

Original author of the Bitcoin system saw the possible issue of attacking the network by accumulating more computing power than honest nodes mining for rewards and showed that it becomes increasingly more difficult as blocks are added and network expands already in his original Bitcoin paper. For this attack to succeed attackers would have to generate the longest chain and to do that they would have to be responsible for majority of CPU power devoted to hashing in the network. This is because someone wanting to modify past block would have to redo the proof-of-work of the block and all blocks added after it before catching up and passing the work of the honest nodes combined[4]. The nodes in Bitcoin network are therefore in a way policing each other. As the input of CPU power that is put into mining is rewarded and as long as playing by the rules is more profitable than attacking the network which is also a lot harder the Bitcoin system is well secured. Even creators and administrators of malicious programs and their networks are known to mine Bitcoins with their botnets rather than trying to attack or deceive the system[13].

The unit of Bitcoin system is 1 BTC. The money supply as well as reward from Bitcoin mining is constantly decreasing. Currently the reward for discovering a block and adding it to the chain is 50 BTC. It will be 25 some time in 2013 and is halved approximately every 4 years until no more coins will enter circulation. There will be a total of almost 21 million of them[14]. This is hard-coded into original Bitcoin software and creates high certainty on the supply. Total amount of Bitcoins in relation to time is shown in figure 2. Nobody can randomly create extra money out of nothing, proof-of-work is needed. Bitcoins are divisible down to eight decimal places so the range of numbers is huge, making dealing with possible deflation resulted from decreasing money supply in case of mass adoption in the future possible[15].

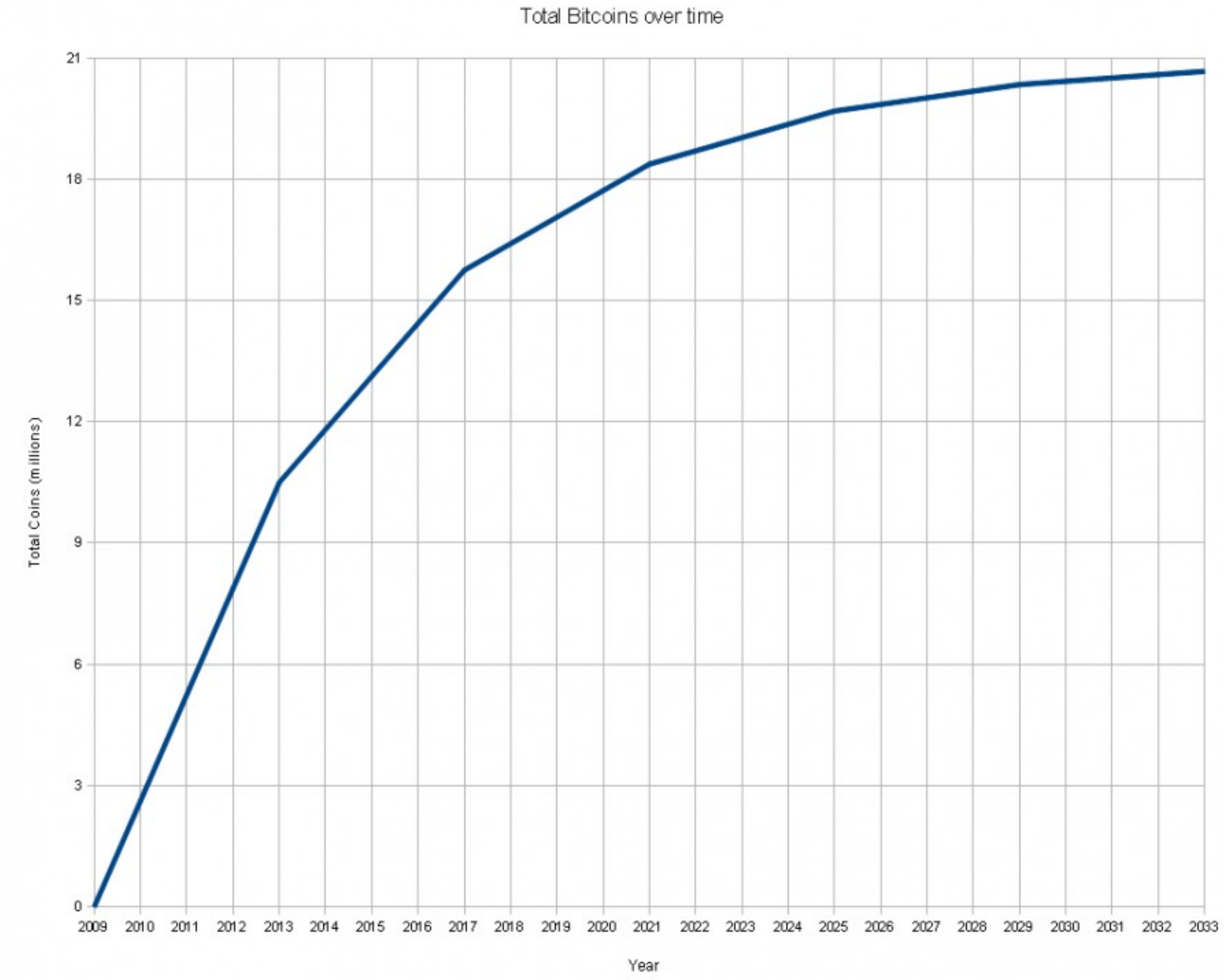

Figure 2: Bitcoin supply [14]

2.4. Transactions

Bitcoin is defined as a chain of digital signatures. Transfer of the coin from one owner to the next starts with digitally signing the public key of new owner and a hash of the previous transaction where the same coins were used. The signatures are added to the coin and the chain of ownership can be verified by checking those signatures[4].

Movement of Bitcoins from one holder to another happen in the network by transferring the value received from transactions inputs to its outputs. One transaction can have several inputs and outputs. Special cases of transfers are redemptions of mining rewards by people successfully generating new blocks, those are transactions with no input[3]. In other cases input is basically an output of previous transaction associated with the coin. To make those links in block-chain visible and verifiable, inputs contain hashes of previous transactions with scripts that contain public key of the receiving party and digital signature of the sender. Public key or Bitcoin address owners will be announced as the new holders of the Bitcoins and they are given the right to redeem the outputs and thus use them as inputs in other transactions. The signature is used to sign the hash of the transaction by the previous owner with their private key. This helps prove that the money was indeed sent by the real holder of the address where the coins come from.

An input claims previous outputs full value. An output however must not claim all the Bitcoins available by inputs. Some of the value may be left unredeemed and is noted as transaction fee which is collected by miner who generates the block that adds the transaction into block-chain[3]. When inputs total value to transaction exceeds the needed amount a special output is created to add a change and keep some of the value to the paying party in given transaction. This works much like Euro payments where buying an item worth 5.59€ and paying with a 2€ coin and a 5€ note a customer receives back 1.41€, the difference is however that now the customer would hold a Bitcoin valued 1.41 if the same transaction would have been carried out in Bitcoins whereas in case of Euros the change would be payed in different nominations of coins.

An example transaction is given in figure 3. First the recipient gives sender their address and asks for 100 BTC for some goods or services. The sender has 3 addresses that have been signed funds to for which he holds private keys and can therefore sign over the outputs from previous transactions now using them as inputs for this transaction. The outputs he can claim and sign over to recipient are for total amount of 105 BTC. Sender decides to leave associated fee worth 1 BTC and signs rest of the balances over the original 100 sent to recipients address back to an address in their own wallet. This results in change from the transaction and there is now another address that has coins they can redeem and use in payments as their wallet contains also the private key for this address.

Now the transaction is sent to network and propagated by all nodes. They add it to their memory as unconfirmed transaction until they see a block sent to them that has this transaction added to it. A miner who has found a solution for this block also collects this 1 BTC fee with other fees and the reward for block discovery. By adding this transaction to block a miner checks the transaction and confirms its valid. Now every other block added to the chain after this block raises the confidence level that the transaction can not be reversed. All this is of course handled by software.

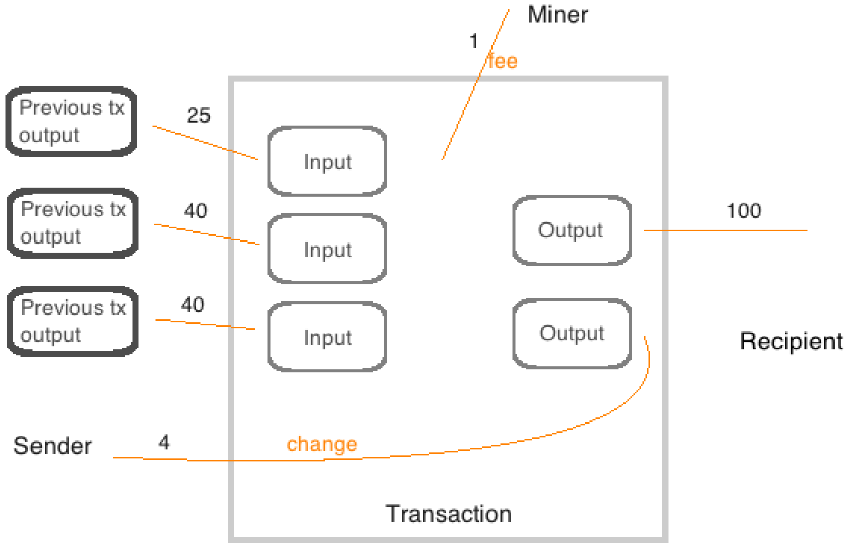

Figure 3: Example transaction

For a transaction to be valid, its inputs total value must be bigger than its outputs with difference added as fees. The spender of the coins must show title to each input used in the transaction. This is tested by evaluating the input script: a description how the owner of the coins can access the coins from outputs from other transactions that are now used as inputs[3]. Typically the script shows the public key of the person receiving funds as an address and signature of person now wanting to spend the coins. Digital signature over the transaction hash proves that spender holds the private key for the key pair where public key shows up in scripts of previous transactions input that the transaction references. Testing the scripts means checking that all the inputs used have the spenders public key in scripts and previous owners signature over the transaction hash is valid.

Accounting techniques are needed in digital currency systems to store and manage rights over time. Those concepts help building complex systems where it is guaranteed that value is not lost as long as everyone follows the rules and it must be possible to easily check where rules are not followed[16]. According to Ian Grigg Bitcoin may have shown the first successful wide scale implementation of triple entry bookkeeping and therefore it is in accounting where Bitcoin has its greatest impact in designing digital currency systems. A triple entry transaction is a 3 party one in which each transaction is digitally signed by multiple parties, including at least one independent. This simple idea is somewhat revolutionary to accounting and big improvement on double entry bookkeeping which has been used for more than 500 years[17].

In Bitcoin all nodes in the network have information on all transactions and miners hash the transactions into blocks. Transactions get confirmed only after they have been included in a block-chain and thus become acknowledged in a collectively maintained timestamped list of all known transactions. The confirmation level depends on number of blocks added to the block-chain after the transaction as each one makes it less possible that added block is not a valid one or part of main chain that all users trust. This means that third party in Bitcoins bookkeeping scheme are miners as well as all the Bitcoin users.

Once the miner has added a transaction into a block it is increasingly difficult for someone to modify it because they would have to regenerate all the blocks after the transaction, making double-spending and reversing of previous transactions practically impossible. That means that Bitcoin transactions quickly become irreversible as new blocks are added to block-chain and there are no chargebacks[3]. This is one of Bitcoins advantages over credit cards: merchants have greater certainty that the funds they receive are final and nobody can reverse the payments. This is why the fees are also considerably higher for credit card transfers than for Bitcoins. For cards honest customers also pay for fraudulent activities online as credit card fraud is very wide-spread.

Credit card data may also be lost to hackers who gain access to databases holding the numbers, sniff the traffic with transaction information or use phising methods to lure people into giving them their credit card details. Those are the problems that Bitcoin users do not have to deal with. The private keys are never moved across the network, they are not managed in central servers and casual users do not even know how to accidentally give out their private keys while knowledgeable users are unlikely to fall as a victim of such an attack. This makes phising attacks against Bitcoin users pointless. Bitcoin users are not however totally safe from hackers, since there is specific malware targeting them, that tries to get hold of their wallet files with keys to spend their coins. One of the users lost 25 000 BTC to a hacker from his machine, amount worth about $500 000 at exchange rates at the time of theft in June 2011[18].

2.5. Bitcoin history

Before looking at how Bitcoin came to life, how it has done during its short existence and what works have influenced the creator of the system lets ask why do we need virtual currencies, why do we need Bitcoin?

Famous economist Milton Friedman has suggested using automated system that would increase the money supply at a steady predetermined rate to control inflation and give more certainty in making spending and investment decisions. He even hinted abolishing the Federal Reserve, central banking system of the United States[19]. While Bitcoin is unlikely candidate for becoming a national currency for a country any time soon it does exactly what Friedman wished for: thanks to mining, changing difficulty of this process and its rewarding scheme Bitcoin indeed gives a well defined and transparent money supply.

Besides controlled money supply and therefore putting a lid on inflation Bitcoin offers several other strengths in both overall design of the system as well as useful properties for end users. Peer-to-peer design allows for fast international transfers from any country with very low fees and running costs. Bitcoin economy works constantly, there are no holidays, there is no need to trust any central organization with ones funds and nobody can seize Bitcoin users accounts.

Bitcoin offers more security than credit cards. The risk of identity theft is mitigated, users personal information is not sent across networks. Fraud is not a big factor with Bitcoins, counterfeit Bitcoins can not be produced. Bitcoin is also easy to use for transactions. Both clients and merchants can start using it with little efforts. Even though thorough understanding of Bitcoins needs some background and basic knowledge of cryptography most users only need to know how to use the software. In general transactions in Bitcoins are easier than a lot of other online payment systems because there is only one identifier for sending and receiving coins: the public key of signing key pair that acts as Bitcoin address, there are no accounts or login credentials.

Bitcoin was introduced by Satoshi Nakomoto, who published his paper “Bitcoin: A Peer-to-Peer Electronic Cash System”[4] in cryptography mailing list in 2008. First piece of client software was released in January 9th 2009[20]. Nakamoto is most likely not a real person, its a pseudonym used by very clever cryptographer and programmer, who wishes to stay anonymous. It is also possible that a group of people are behind this name. Nakamoto claimes to be 36-year-old Japanese who spent more than a year writing the software driven by anger about worldwide financial crises. Joshua Davis of The New Yorker questions the fact that Nakamoto is from Japan as he usually posted to Bitcoin forums after British work hours and had perfect command of written English[21].

Nakamotos real name is not very important and it is understandable why he wishes to stay anonymous after creating a currency that catches a huge amount of negative attention from governments not only due to its possible uses in illegal activities but also because it can create untraceable alternative economy that may function without paying taxes to countries where business is conducted or even jeopardize use of currencies backed and controlled by governments.

Nakamotos work might become revolutionary breakthrough in financial world after many commercial ventures to popularize digital currencies like David Chaums Digicash have failed[22] due to not having one of the layers of Financial Cryptography properly handled. The 7 layers of problems that need to be dealt with for such systems to succeed are cryptography, software engineering, handling rights, dealing with accounting and governance, answering how the instruments carry monetary value and how the previous layers can be used to have financial meaning according to Ian Grigg[16]. Those layers are built on top of each other and when one of them fails the system collapses. Digicash had the lower levels with mathematical solutions handled pretty well but lacked on management side of the business, point where Bitcoin cannot fail since it has no central company to manage it. It is also possible that world at large was not yet ready for such a currency in 1994 with technology and mindset of people only starting to shape into accepting information society where we live today.

Bitcoin uses several ideas and works besides Digicash to learn from. One of the most notable contribution was made by Wei Dai who proposed a system where no-one needs to be trusted, where all participants have information of all transactions and money is created by solving computational problems in theoretical design paper called “b-money” in 1998[23]. From this Bitcoin draws several of its most important ideas how to solve both technical and management problems for creating digital currencies such as block-chain for hashing the transactions into a hash-linked chain via mining and signing the transactions with private keys where public keys are addresses of receiving parties in those contracts.

Proof-of-work concept was added by Adam Back who first proposed combining his hashcash idea that helps fight spam and denial of service attacks by making them computationally more expensive with Wei Dais b-money idea by using hashcash as a minting mechanism for digital currencies with no central authority in 2002[24]. Bitcoin offers a reward to miners who find a nonce that together with all transactions and reference to prior block can be hashed together to create a block so that it would match requirements set up by the network. As finding such a nonce is difficult and all participants in Bitcoin can see the difficulty of given block creation there is transparent proof-of-work.

The volatility of Bitcoin has been huge during its short existence. After getting a lot of media attention in the spring of 2011 the price of Bitcoin started rising quickly. From early April to the end of May, the going rate for 1 Bitcoin rose from 86 cents to $8,89. In June 2011 Gawker published an article about currency’s popularity among online drug dealers, showing that all kinds of illegal drugs and pharmaceuticals can be purchased for Bitcoin on marketplace called the Silk Road. Silk Road runs as hidden online service and is accessible only using Tor, a system that routes users traffic through several servers encrypting it in the process to conceal communicating parties locations and type of data transferred[25]. Bitcoin value more than tripled in a week as a result, reaching a maximum of almost $30 soon after[26].

Bitcoin bubble exploded as fast as it was created and the fact that Mt. Gox, the site responsible for most of the Bitcoin to USD currency exchange got hacked did not influence the value of Bitcoins positively. Mt. Gox lost its users database with about 60 000 records to a hacker who was most likely from Taiwan. The passwords in the database were hashed with MD5, but hacker was able to find corresponding plain text passwords for several users using rainbow tables or brute forcing and log into their accounts as a result. Those users lost money on their accounts, but hacker did not gain more than $1000 due to fast reactions by the Mt. Gox team and limits they had put up. The damage to Bitcoin was done however and its value dropped to $0,01 for a while[27]. The hack of the biggest exchange was the breaking point for Bitcoin value, but the burst of the bubble was probably inevitable.

During the last few months Bitcoins exchange rate to real-world currencies has been more stable however, staying around $5 mark for a while now but there is no certainty that times with high volatility are over for Bitcoin and value is given to it by normal users not speculators. There is always the possibility though that supply and demand have finally shaken the Bitcoin exchange rates to currencies like Euro or Dollar into place and the roller coaster days for market graphs seen in figure 4 are over.

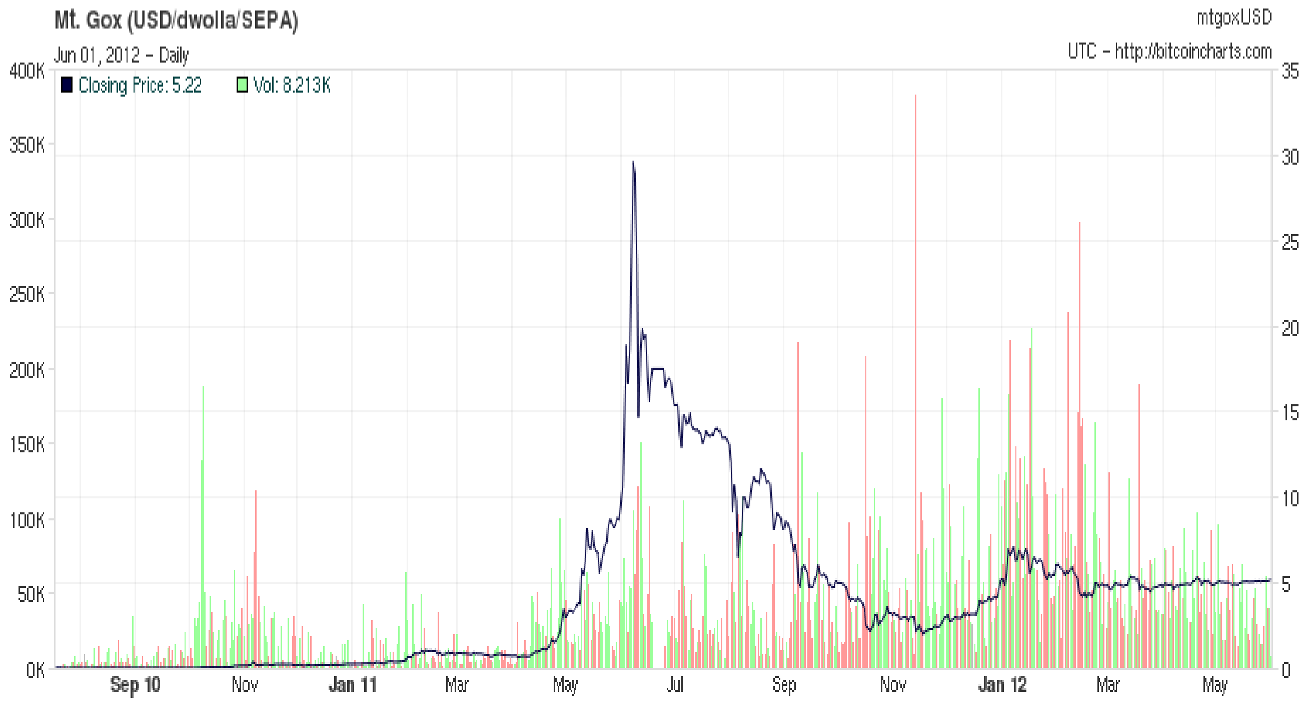

Figure 4: Historic Bitcoin to US Dollar price (source bitcoincharts.com)

Total number of businesses willing to trade in Bitcoins is still pretty limited and even more so when you look only for legitimate possibilities that are not looked wrongly upon by governments. Besides illegal drug trade several shadowy organizations accept Bitcoins as donations. They include international hacking group LulzSec that is responsible for several high profile attacks like compromising Sony user accounts in 2011 and Wikileaks, a website publicizing private, secret and classified information[28]. Besides the mentioned parties and other Bitcoin uses not in favor of governments one may buy music, clothing, electronics, web hosting, pay for professional services, accommodation or their bar tabs via Bitcoin. There are thousands of possibilities and accepting Bitcoins has been made very easy for merchants. So the wrong kind of attention brought on to the currency is just something that comes along with decentralization and fair amount of anonymity that Bitcoin traders have.

It is pretty impossible to determine how big is the legitimate Bitcoin economy compared to illegitimate uses, but we can accurately measure the amount of transactions in Bitcoin system as this is all public information and therefore we can know the approximate size of Bitcoin economy and its trends. Figure 5 shows average amount of transactions per day during the last year. In April 2012 there were about 7000 to 8000 transactions per day on average[29] and with those figures Bitcoin is far from competing with credit cards or popular e-wallet systems. Popularity of the system is on the rise in general however, much of the growth in amount of transactions to 13 000 a day in early May comes from increase of activity in transparent Bitcoin casino called Satoshi Dice that offers more than 99% breakeven odds with instant payouts on winnings[30].

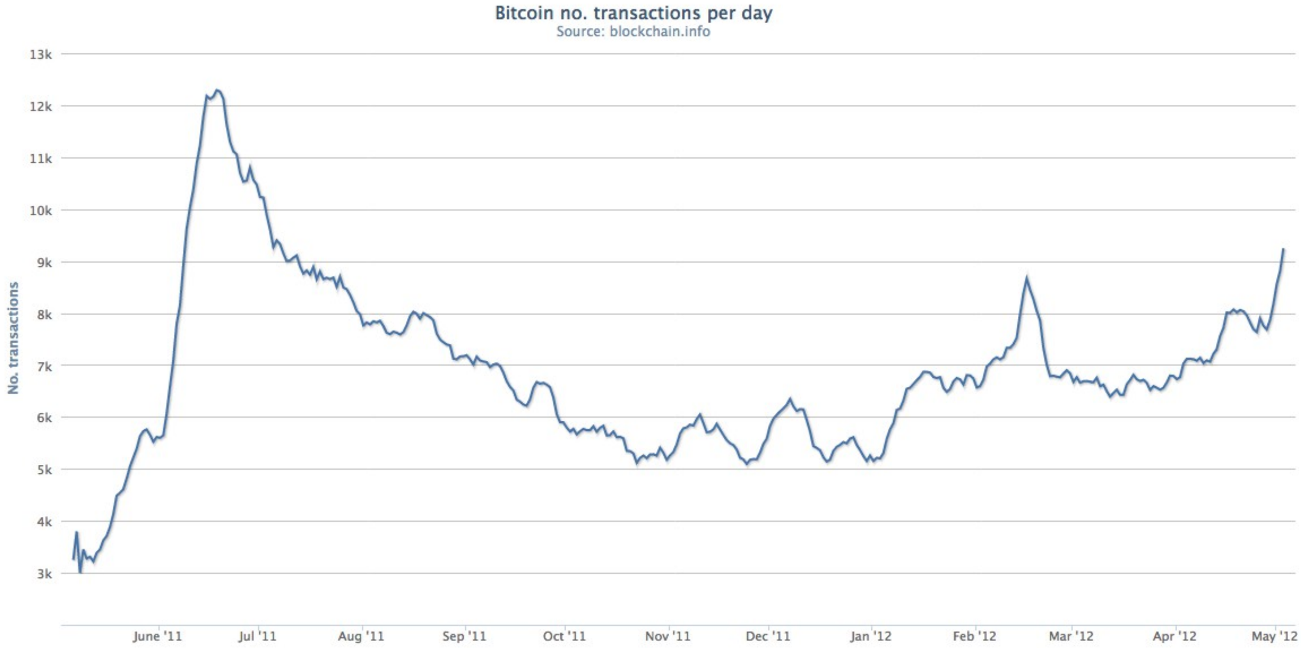

Figure 5: Number of Bitcoin transactions per day (source blockchain.info)

Transactions per day can be somewhat inaccurate measure of activity in Bitcoin economy because it also shows coins sent back and forth between addresses owned by same person. To show a genuine level of economic activity a metric called Bitcoin Days Destroyed is often used. This takes into account the amount of coins sent and multiplies it with the days they have remained unspent thus it is not giving weight to small transactions or dealings with coins that are regularly spent. Bitcoin Days Destroyed is constantly growing however[31] and in reality is not a lot better indicator for Bitcoin. The growth comes from the fact that as Bitcoin matures there are more and more older coins that have been unspent for a while and also because mining grows the overall number of coins in circulation constantly.

Once we have a working decentralized cryptographic currency system one may ask why should they give any value to Bitcoin? Why do Bitcoins have a price in dollars or euros if its not backed by authoritative entity? Besides properties given to Bitcoin because of its peer-to-peer design and interesting use of cryptography Erik Voorhees finds currency to be valuable because of its usefulness and scarcity. Bitcoin scarcity is predetermined: there will never be more than 21 million Bitcoins in circulation. To see why Bitcoins are useful as money lets compare Bitcoins to gold, that has been widely used as money for centuries. Both gold and Bitcoins are useful as means of exchange because they are scarce, easily identifiable, hard to counterfeit and divisible, their pieces can be combined for transactions, the supplies are rather steady and predictable and they are easily transportable and manageable[9]. If we look at those properties of money we see why both are useful as means of exchange and looking at them one by one Bitcoin outperforms gold in all categories, especially in terms of transportation and storage. Bitcoin certainly has its place in the world of finance making a lot of interesting concepts available in one system but for wide-scale adoption, getting out of proof-of-concept phase and becoming a major player in international economy a big push in amount of use cases and users is needed.

2.6. Cryptography

Cryptography solves mathematical problems coming up when creating digital cash systems. It delivers useful properties like confidentiality, integrity and authenticity. For Bitcoin more important of those are authenticity and integrity. It has to be possible to mathematically prove that sender of the coins is indeed someone who has rights to spend them and that their spending messages are not tampered in the network. According to Ian Grigg's 7 layer model for digital currencies, cryptography is in the bottom level, it is the basis of all such systems. Properties available thanks to cryptography are used by disciplines on higher layers of the model to create a secure system[16]. If cryptography fails in any major way the system collapses.

Several security experts like Dan Kaminsky have audited the system and the code of the original client software and found that Bitcoins use of cryptography is unorthodox. Kaminsky noted that Bitcoin is really well designed and interesting solutions in cryptography work together to offer security good enough to be used in peer-to-peer network structure dealing with money[32].

Bitcoin is based on public-key cryptography using Elliptic Curve Digital Signature Algorithm (ECDSA). Well known hashing algorithms SHA-256 and RIPEMD-160 are used to compute Bitcoin addresses and for linking both transactions as well as blocks to each other. Block-chain is basically a hash chain based on time-stamping and the way Bitcoin transactions are linked together and added into blocks resembles a Merkle tree.

ECDSA is a variant of digital signature algorithm that uses public-key cryptography based on elliptic curves over finite fields. The algorithm is named like this since the curves used to calculate the keys are described by cubic equations that are similar to finding circumference of an ellipse, but they do not represent ellipses. Equation for such curve can be described in a formula

$$y^2 = x^3 + ax + b \quad [33]$$

This is a cubic equation since the highest component it contains is 3. To plot such a curve component y can be calculated when you know component x and given values of a and b. For simplified explanation we are excluding other mathematical parameters like points on the curve and prime modulus that is used to limit the length of keys and only use one reference point called G which defines the curve. When G is multiplied with a random number we use as private key we get another point on the curve we now use as the public key we can share with others. Hash of the message signed and private key are used to calculate the signature that can be verified by checking it against the public key using the mathematic properties offered by elliptic curves[65].

Most digital signature algorithms use RSA that is based on difficulty of factoring large integers. The key lengths for secure RSA have increased however and it is crucial to keep the size of messages that are sent over the network in peer-to-peer systems under control. ECDSA offers equal security for a far smaller key size, reducing the processing overhead and amount of data needed to be stored and transferred across network. On the other hand no mathematical proof of security has been published and it has not been in wider use as long as algorithms that use RSA and therefore the confidence level is not as high[33].

The most important hashing algorithm used in Bitcoin is SHA-256. Secure Hashing Algorithm is a cryptographic function that takes in a block of data of arbitrary length and always returns a 256 bit string so that finding the message from the digest returned after calculations within the function is almost impossible and when a message in data is changed the hash value most probably changes drastically. The hash is also easily computable and it is infeasible to find 2 messages with the same hash, a result of which would be called collision. For SHA-256 there are in fact no collisions found[33].

Finding a SHA-256 hash of data is a process of doing bitwise operations on the message that is padded to have a length that is modulo 512. Operations such as integer addition, bitwise and, exclusive or and or, logical shift right and bit rotation are performed in 64 consecutive rounds modifying the message in 512 bit chunks[34].

In a Bitcoins block-chain every block has a header. This header contains the version number, hash of the previous block, Merkle root, timestamp, mining difficulty and a nonce[35]. If the block header is hashed so that the result satisfies the difficulty at the given time a new block is added to the chain and the adding miner gets a reward as discussed in Mining section. For finding such a hash the nonces integer value is incremented constantly.

Merkle root in the header is a cumulative hash of all transactions starting with a block generation reward sent to miners address. This means that all miners start looking for nonce that would result in a hash that meets the difficulty with different block headers. While a miner tries solving the block new transactions are constantly added as new valid transactions are received from other nodes in the network thus changing the Merkle tree and its root hash. The timestamp in block header is kept up to date while mining.

Merkle root is like a summary of all transactions in a given block. It is found by hashing transactions into Merkle tree. Merkle tree is a binary tree where the root is found by hashing together data by 2 consecutive pieces of information and then doing same on next level with the resulting digests from previous round until 1 cumulative hash is produced. If one of the nodes in the tree does not have a pairing value its concatenated with itself before tree is expanded so that for all intermediate nodes there are 2 input hashes. Hashing together 9 Bitcoin transactions would result in a tree that goes 4 levels deep and last level would have this blocks reward transaction to eight transaction hashed into Merkle tree and then hashed together with ninth transactions hash concatenated with itself to result in Merkle root for this block. Graphical representation of a block and Merkle tree of transactions in it is presented in figure 6.

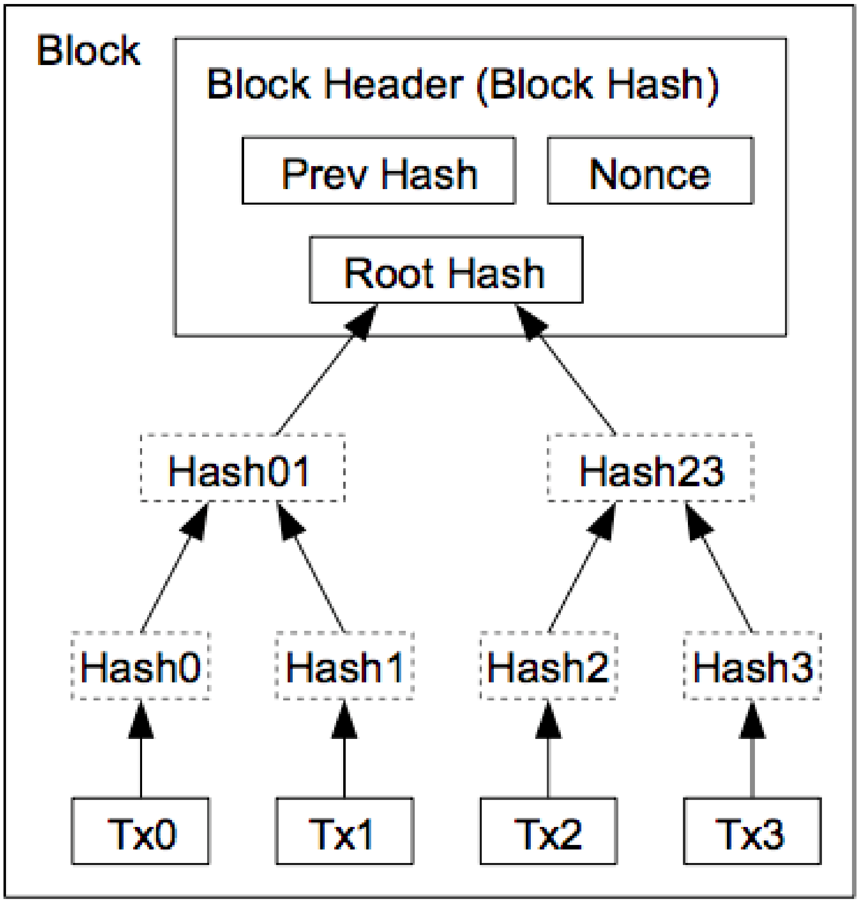

Figure 6: Bitcoin block [4]

The blocks form a timestamped linear chain known as block-chain in Bitcoin as a result of mining. In this chain every block except the first one holds a reference to chronologically previous block and its block headers solved hash is referenced by the next block after it. For both mining and building the Merkle tree from the transaction hashes within the block double SHA-256 is used and its result is interpreted as little-endian number. Little-endian means that by value the hash starts with the least significant components. This lets the mining hashes start with leading zeros to meet the difficulty of mining.

3. Security of Bitcoin

3.1. Breaking the cryptography

Bitcoins strength is that it uses cryptography in a way no other system before it and actually makes it work. It is a currency that does not need central party to manage it, everything is defined by laws of mathematics. But as Bruce Schneier puts it cryptographic system can only be as strong as the algorithms it relies on and when any of them is broken the system goes down[36]. This is especially true for Bitcoin since it is a system heavily built on cryptographic knowledge. Failure of algorithms for Bitcoin would mean one of the main cryptographic systems used to be broken. Those are ECDSA, SHA-256 and RIPEMD-160. All are published algorithms with quite a bit of research going into them. How can someone break cryptographic algorithms and what happens with Bitcoin if one of the important algorithms is broken?

3.1.1. SHA-256 collisions

SHA-256 or other hashing algorithms have two different attacks we should worry about: collision and preimage attack. Collision is a situation where different inputs are hashed into same digest value. Finding a collision for a SHA-256 via brute force attack is possible since it has limited amount of different hash values it can produce. There are a total of 2^{256} results for hashing so collisions are very unlikely to happen and we are not concerned about such a small possibility. On average a good attacker using the birthday paradox to their advantage is likely to find a collision in “only” 2^{128} tries for SHA-256 and we need a lot better chances of finding a collision to consider an algorithm broken. If there is an easier method found for looking for collisions than brute-forcing due to cryptanalysis it is considered that there is a flaw in the algorithm[37].

In 2005 Chinese cryptographers broke SHA-1: they developed a method for finding collisions 2000 times faster than brute-forcing[38]. Their method has since been outperformed by other cryptographers work and machines have become a lot more powerful in last 7 years but finding a collision would still take a lot of computing resources and luck. If we theoretically think of a crypto-currency system similar to Bitcoin but developed before 2005 and using SHA-1 as main hashing algorithm, what would breaking of the function mean to the system 7 years after publishing the first paper on how to find collisions faster than brute-forcing?

First of all Bitcoin would not be theoretically secure if it used SHA-1, but attacks would still not relate well to practice and finding exploitable holes in a system would not be easy. In Bitcoin hashing is used most importantly in mining and transactions. For transactions it is a matter of signing the hash of the transaction to transfer the value of the coins to another user. If someone was able to find a way to create a transaction that would result in the same hash value as the original one they would be able to steal the coins by adding themselves as a receiver of the coins. The original senders signature over the transactions hash would be valid and therefore also the transaction would look like a valid one. There are however limitations to this clever attack. Attacker would have to find this very specific collision instead of just a collision: a transaction message that has their Bitcoin address instead of intended recipients address in it would have to have the same hash. For this weaknesses found in SHA-1 are by far not enough. In addition the attacker has to be quicker than owner of the coins in spending them.

After the transaction is added to a block by miner the attacker would be able to use outputs of the previous transactions inputs and spend the coins. Both transaction messages would reference same previous transaction with different scripts added with different owners having the rights to spend the coins by using the outputs of the previous transaction. If the attacker finds the colliding transaction hash after the original owner has spent the outputs his efforts would become useless.

As there is a remote possibility that several transactions may hash into same digest in the future a mitigation is developed well before collisions in SHA-256 make such an attack feasible. Standard Bitcoin client does not add transactions to database if they see them coming in with a hash that is already saved there. It takes the transaction coming in as a duplicate and discards it as shown in Appendix B. For the protocol in general however this might become a problem if some people start using the thin clients that do not keep all the transactions in their database.

Attacker would want to find collisions in block hashes to steal the transaction fees and block discovery bonus or invalidate some or all transactions for denial-of-service or double-spending attack. How possible collisions affect mining process and integrity of the block-chain? Unlike transactions blocks do not live on their own. There can be two transactions with same hash value in the block-chain and both can reference same previous transaction due to collision. For blocks however this is not possible since they form a timestamped linear hash chain. Attackers block that has same hash value as one of the previous blocks would not be added into the chain if they referenced the same previous block as the block they want to replace. Every new block discovered has previous blocks hash and timestamp in its header and blocks have to be in chronological order. Standard Bitcoin client does not accept a block that has a hash that is previously saved in the database, code for this is shown in Appendix C.

Ahto Buldas and Sven Laur showed that for building a secure timestamping service the hashing functions used on the server side do not need to be collision resistant, preimage resistant and not even only one-way[39]. This means that in terms of integrity of the block-chain breaking the hashing algorithm of SHA-256 has no real effect. Old chain would stay unharmed with all the transactions hashed in there when change of hashing algorithm would be needed for other reasons discussed in this chapter. Then the hashing would continue with last block solved with old hash as input reference point to new in the block that is agreed upon by the community as the starting point of new hashing algorithm for mining process.

Another cryptographic attack theoretically possible is improving the hashing algorithm for SHA-256. If someone found a way to find double SHA-256 of blocks headers significantly faster than others they would gain an edge in mining. They could gain a monopoly in adding blocks to the block-chain with help of large amount of computing power and reverse their own transactions or use it for denial-of-service against miners and regular users by building empty blocks and not including transactions. The effects of this are discussed in more detail in next chapter. The improvement of SHA-256 hashing algorithm would have possible effect on Bitcoin only if the improvements stay private. If many miners start using more efficient algorithms mining difficulty would increase and system would continue functioning as normal.

3.1.2. Attacking transaction signatures

We also have to look into possible collisions happening in RIPEMD-160. Those are 2^{96} times more likely to happen than collisions in SHA-256 since the hash length is 160 bits instead of 256 bits. RIPEMD-160 is used for creating Bitcoin addresses which are used to identify where coins are sent. This means that if someone finds a ECDSA key pair where public key would hash into same RIPEMD-160 digest value as another persons Bitcoin address he would be able to spend all the coins that this address holds. But to create this kind of a collision attacker would have to find a valid ECDSA key pair that would hash into colliding RIPEMD-160 hash value and the process of hashing a public key into an address involves first using SHA-256 and then RIPEMD-160 before calculating a checksum with double SHA-256 and encoding it to a Bitcoin address[7].

In 2006 a team in Graz University of Technology showed that methods used to find collisions in SHA-1 or RIPEMD did not extend against RIPEMD-160 and the algorithm was secure to known attacks[40]. This means that only attack method would be brute-forcing which would consist of generating ECDSA key pairs before hashing them with SHA-256 and RIPEMD-160. So in theory using RIPEMD-160 makes Bitcoin protocol less secure with offering shortening of the public keys to be conveniently useable as addresses due to its shorter hash length but in reality process of finding collisions involves too many rounds of calculations to make such an attack feasible.

Then there are attacks on ECDSA. If someone was able to find a way to calculate private keys for key pairs where corresponding Bitcoin address has funds sent to it they would be able to spend it as having the private key is all one needs to sign transaction messages and pass on the value. Private key is a 256-bit integer[41] therefore having 2^{256} different values and with this it is more resistant to brute-forcing than Bitcoin address created from the public key due to hashing it with RIPEMD-160 that has 2^{160} different values meaning that on average one Bitcoin addresses balances can be redeemed with 2^{96} different key pairs. Extra computational difficulty behind it is discussed in previous paragraph: one would have to compute a public key for private key, then hash it twice with 2 different algorithms.

Lets look at chances for an attacker trying to find RIPEMD-160 hash-value that collides with another Bitcoin address to be able to spend the coins. On average finding a collision would take minimum of 2^{80} tries of hashing. Lets say that an attacker has same amount of computing power as all the miners currently trying to solve a block which is about 12 terahashes ($12 * 10^{12}$ hashes) per second[11]. Total computing power of the Bitcoin system is calculated by mining output and within 1 mining output hash SHA-256 is calculated twice as in most cases of the usage of the algorithm within Bitcoin takes double SHA-256 of the input. Lets generously say that SHA-256 hashing in mining process takes the same amount of time that the whole computing difficulty behind generating an address off of a private key would take for an attacker. We will see that on average the attacker succeeds in $2^{80} / 10^{12}$ seconds, which is more than 38000 years. We are going to have to take into account that computing power increases over time. Lets have it double every 18 months as has been often quoted version of the Moore's law[42]. Now we will be able to find a private key in about 16,5 years as shown with Appendix A. This is more than 16 years of constant hashing with highly optimistic estimations to find private key behind 1 particular Bitcoin address. As we stand today brute-forcing is infeasible, but we have to keep an eye on developments in cryptography, computing power and perhaps even quantum computing and be able to make adjustments in algorithms used in the system.

If tackling ECDSA keys by brute-forcing is unfeasible we have to find a better method to attack Bitcoins signature algorithm. Bitcoin uses secp256k1 elliptic curve that has 256-bit private key and is based on Koblitz Curve[43]. Algorithms using Koblitz Curves are not part of National Security Agency, ANSI or other standards and therefore not researched and analyzed as extensively as some of the other ECDSA-s. Therefore it can be considered less secure and Fabio Pietrosanti suggests avoiding such an algorithm for those reasons[44]. Bitcoin seams to be the only widely used system that uses ECDSA based on Koblitz Curve and it looks like this is the part where author of Bitcoin may have not made the best choice, choosing speed over security. At the same time no weaknesses are published for ECDSA and keys are hidden behind hashing algorithms.

Lets assume someone found actual weakness in ECDSA implemented in Bitcoin and was able to crack the algorithm and trivially find private keys from public keys. Now attackers would be able to forge signatures and therefore sign transaction messages with displacing coins they do not own. But for Bitcoin attackers would not be able to get to those keys to steal users money since they would first have to get the public keys to start calculating private keys of those key pairs. The public keys however are hashed in the system. Successful preimage attack on both RIPEMD-160 and SHA-256 is needed before it is possible to use any of the weaknesses found in ECDSA because the public keys themselves are not broadcast in the network before the coins are signed over to next party and therefore spent[45]. Only Bitcoin addresses are available to attackers for most addresses and those are built by first hashing the public key with SHA-256 and then with RIPEMD-160. This means that only addresses that are reused are subject of this attack as they have revealed their public keys but this is currently not an issue since no weaknesses in ECDSA are known and users can increase their security and anonymity by using different addresses for all transactions.

3.1.3. Preimage attack

Preimage attack on a hash function means looking for the original message from the hash value produced by the hashing calculations. Besides mandatory execution of preimage attack to find private keys preimage would also help attackers to mine coins faster. If they found a way to get a nonce from any of the hashes that meet the difficulty required for a given block they could present it as proof-of-work while collecting fees and discovery bonus for finding a new block and adding it to the chain. This kind of preimage attack would be interesting one as there are several hashes that can be attacked and attacker can also control part of the message that is going to be hashed. Attacker can change the Merkle root by deciding which transactions are added to the block he is trying to hash and to which address the reward is sent. At the same time he is only interested in finding the integer value of the nonce.

Currently best preimage attack for SHA-256 is against 41-step version of the hashing algorithm. The 64-step process is still secure against this meet-in-the-middle attack[46]. Meet-in-the-middle attack means attacking the cryptography of a hash function by working from both ends of the hash at the same time. It tries taking the possible message values closer to the hash digest while taking hash values closer to the original message until they meet in the middle and reveal the input of hashing. In principle this is exactly the kind of attack that could succeed for Bitcoin as finding a suitable nonce is meeting in the middle. Added difficulty in launching a preimage attack is brought on by the fact that block headers use double SHA-256, but at the same time a preimage found does not have to be specific: any nonce helping to hash into any of the acceptable hash values is sufficient. This is one of the attacks needing more research as there may be Bitcoin specific meet-in-the-middle attacks possible against double SHA-256. If someone found a method for this it is pretty likely that they would not publish it however as even a small edge found in mining is valuable.

Currently Bitcoins cryptography is very strong: brute-forcing is infeasible, algorithms are strong and in case of weakening of algorithms several mitigations are already in place beforehand. With developments in cryptanalysis and computational speeds however longer key-sizes and hash lengths or better algorithms have to be implemented in Bitcoin in the future. Although the creator of the system has announced the possibility of changing cryptographic algorithms in the system seamlessly for users in the unlikely occasion that SHA-256 gets broken any time soon[47] there is no concrete plan for doing so.

3.2. Attacking with computing power

Bitcoin fights double-spending by adding all broadcast transactions into block-chain. Block-chain is the database of all transactions and the branch of the chain that has the highest computational cost associated with it is trusted by nodes in peer-to-peer network[48]. Honest miners build on top of the longest valid chain. They are rewarded Bitcoins for doing so and in case they would on purpose or by accident add blocks to the chain that is not considered as main branch by the network the coins they received by claiming the block discovery bonus and transaction fees would not be spendable as they are not included in the trusted chain. Clients should also trust only the transactions included and confirmed by several blocks added to the chain after it so that there is strong evidence that they are part of the main chain and not one of the orphaned chains that are not built on top of blocks that carry the highest amount of computations with them.

3.2.1. Double-spending attack

Branching of the block-chain can happen on purpose in case of attacking but also by a chance when several new blocks are discovered and broadcast to network a few seconds apart. When this happens the nodes in the network generating the blocks start building on top of the block they received first. Now the block that gets referenced by another new block first will become part of main chain and all others will stay as orphans since there is more computational effort associated with this branch[3]. Transactions in orphaned chains go back to unconfirmed state and are added by miners building new blocks later on.

The attacker who can produce a block-chain for which they show proof-of-work, difficulty level matching the hashing speeds and bigger amount of total computational effort than the builders of the main chain would have control over the whole Bitcoin network. If an attacker is able to build such a chain and broadcast the built chain it would be accepted by the network as the main branch of transactions database. The transactions that are included in the previous main branch and not in the one created by the attacker are no longer confirmed by being added in a block by a miner and therefore not trusted.

As an effect of building a new main branch for the block-chain attacker could reverse the transactions he signed and that were added in the previous main branch back until the point where attacker split the chain[49]. The attacker does that by simply not adding the transactions into newly built branch and possibly using the same coins to issue other transactions effectively double-spending them as a result. For the network that trusts the chain with highest computational costs associated the older transaction with the same coins never existed and the poor receiver of the coins in the transaction that now never gets confirmations loses their rights to use them as his transaction is never added to the chain again since the coins are already used and value is signed on to another party instead of him.

The attacker could not however reverse transactions that are not sent by him as he does not know the private keys with which he signs the value over to other parties. He would also not be able to create value out of thin air, proof-of-work and difficulty rules building the blocks have to be followed even when creating an alternative attacking branch of block-chain, otherwise it is not accepted by other nodes. Attacker can not take other peoples money as any of the transactions he would add into blocks created that were not validly signed would not be accepted for payment by other nodes in the network. Furthermore those invalid transactions added to the block would result in the block becoming unacceptable as well[4].

3.2.2. Denial-of-service with computing power

What an attacker may do is not include the transactions in his branch. Those transactions would wait unconfirmed until added to the block-chain later on. This could happen after attacker loses the majority of the computing power in the network, stops his attacking efforts or starts adding transactions by others in the transactions database built. Then the transactions would get the needed level of confirmations to be trusted and transaction would be valid unless someone was able to fork the chain again with their computing power and create another branch after the previous split has become the main chain and before the transaction gets added to a block part of the main chain.

This could result in denial of service. Attackers can choose which transactions are added in the chain. They may in fact add only redemption transaction in their blocks preventing all traffic that passes on value in Bitcoin network making the system worthless. If users can not send and receive payments the currency is very unattractive. This way the attacker also loses out on transaction fees but they may not be concerned about it as their goal in this case is most likely to kill Bitcoins growing popularity and if they keep control long enough they are able to eventually stop using the currency altogether.

The attacker in control also prevents other miners from mining any valid blocks during the time they have the majority of computing power as the other mining effort is put into branch that loses its status of main branch in the block-chain. Smart attacker would build their chain quietly in the background and not broadcast the blocks discovered to the network. They would have to use more computing power than the Bitcoin network combined during this building in the background. Once they unexpectedly to other users make their efforts public their chain is accepted as the primary chain by Bitcoin protocol. If this attack is run over extended periods of time attackers risk loosing out on total processing power if honest nodes have passed it and he is unable to keep up. Then all his efforts will become useless and it is highly unlikely that Bitcoin community will ever know that an attack was launched. At the same time the longer the period of control for the attacker the bigger the damage to Bitcoin. Few hours worth of unconfirmed transactions would not create a chaos but more than a week of financial activity rolled back would make average users lose trust in the system.

3.2.3. Difficulty of attacking with computing power

So how hard it is for an attacker to get and keep the computational effort high enough to build alternative chain that would be accepted as the one valid branch of block-chain? As mentioned earlier Bitcoins current hashrate that all miners combined produce is 12 terahashes every second, which is equivalent to 1,5 * 10^{17} floating point operations per second (FLOPS)[11]. At the same time the power of world's 500 most powerful supercomputers combined is about 4,4 * 10^{16} FLOPS[50]. This means that if someone was able to have those 500 supercomputers mining together at full power they would operate a highly successful mining operation, discovering 23% of the blocks on average, but it would not be enough to control the network.

For attacking with computing power to succeed at least one of two conditions has to be met: a highly motivated attacker with huge amount of resources or decrease in mining activity. The motivated attacker would have to specifically target Bitcoin with most likely destructive objectives instead of financial gain in mind. Parties like alternative currency systems or governments could be behind such attacks for various reasons not discussed here.

The decrease in mining activity in the future is quite possible. Due to the size of reward for discovering a block halves around every 4 years the mining incentive drops as well. At the same time Bitcoin is deflationary due to decreasing money supply. This means that the incentive for theft and profitability of getting massive amounts of computing power for double-spending by creating alternative chain of transactions database rises[3]. So needing less to attack and gaining more if successful could create just the right conditions for attackers.

3.2.4. Mitigations for attacks with computing power

Although ridiculously large numbers are associated with computing power needed for this history revision attack and there are not many parties who could afford such a mission the threat is real and there are not enough mitigations in place. Only real defense against this attack is currently the fact that launching it is highly difficult in terms of computing power needed and honest mining with same power would be more profitable for an attacker.

If someone successfully launched an attack with computing power and was able to keep double the amount of hashing of the whole Bitcoin network combined during next 2 years, they would in theory be able to split alternative branch from the genesis block and present it with higher amount of total computational cost associated thus revise the entire transaction history[3]. In reality rewriting the entire history would not work as there are checkpoints in current main chain hardcoded in the client software. Hashes of blocks that are trusted are added with every new version of Bitcoin software[51]. Attacker would have to fork the chain from the last official checkpoint and merge it back before another trusted checkpoint is added in the software. Discovering such an attack is relatively trivial: if large amount of blocks previously part of main chain are now orphaned and several transactions that were previously confirmed are now unconfirmed then someone most likely successfully launched an attack.

To mitigate attacks with a lot of computing power a combination of 2 kinds of methods could be used: enforcing fees by protocol or using technical means other than hard-coded checkpoints for trusted chain. By idea Bitcoin will be kept secure by miners even if mining rewards diminish because of transaction fees they collect on transactions that are added to the block that they hash and broadcast to network. For this to actually be true Bitcoin popularity would have to keep growing and even then an enforcement on fees could be introduced since mining has to be worthwhile for participants to continuously output big hashing power to make attacking difficult. With voluntary low fees the transaction volume would have to be huge for miners to break even with electricity costs going into constant hashing. This mitigation could be easily introduced by miners themselves however: if some of the bigger pools only start adding transactions to blocks for confirmations after certain percentage for fee is met they enforce the rules on the network.

Bitcoin developers are careful in adding checkpoints, they do not add them to newly discovered blocks that are not yet proven to be part of main chain and therefore this is not a realtime mitigation although it eliminates the chance of rewriting the whole history of transactions with empty database. Better technical mitigations for attacks with computing power that would allow constant control over branching without centralizing the currency are more challenging. In fact the mitigation put in place by the developers is not a decentralized way since the community has to trust the people hard-coding the checkpoints and therefore not part of pure Bitcoin protocol[52].

Extra decentralized peer-to-peer mitigation for attacks with computing power could be introduced by adding automatic checkpoints. If a client thinks that the block is now trusted it takes it as part of main chain even when later on there is a branch that surpasses the total computing power. Clients could rate the incoming packets by combination of metrics including but not limited to the mining difficulty, time the block was received and the discovery timestamps difference from reception, the miner discovering the block, giving credit to entities known to regularly add good blocks to main chain but not having the previous or next block mined, amount of valid transactions hashed into block and rating of the blocks added after the block to the same chain. Similar idea was introduced by Gavin Andersen who expressed that this could be a side system monitoring the block-chain[53].

If implemented in original Bitcoin software the checkpoint adding would use majority voting. If most of the nodes in the network have rated a block high enough to be considered a checkpoint an attacker could no longer branch a chain before this checkpoint since most of the network will not accept his blocks. The mitigation proposed is currently a rough idea and should be investigated further.

3.3. Cancer nodes

Attacking Bitcoin network or targeted users with cancer nodes would mean filling the network with clients controlled by the attacker. The goal for this would be making either a user or users connect only to malicious nodes or separating one part of Bitcoin network from others. As a result of flooding the network with cancer nodes an attacker could refuse relaying blocks and transactions creating denial-of-service. If he is also able to segment the network he may create a condition of several block-chain branches to be built simultaneously having no knowledge of others existence[49].

In case a successful network split by running huge amount of cancer nodes the attacker would be able to double-spend coins similarly to methods discussed in attacks with computing power with lesser effort. They would create a situation where a part of network builds on top of 1 branch and trust the transactions within this chain they think is part of the main branch. In reality after cancer nodes disconnect and the network realizes that there has been a fork in the block-chain and resolves it by choosing to trust the branch with biggest amount of total computing power put into building the blocks within as specified by the protocol. The transactions in now orphaned blocks are left unconfirmed and for some of them the attacker may have been able to spend associated coins in other branch.

In case the network segmentation is not complete, the attack with cancer nodes fails. If the user who attacker wants to disconnect from the network connects to one honest node who is in turn connected to peer-to-peer network by at least one non-malicious node, he gets enough information on transactions and blocks discovered to stay unharmed. This makes the total segmentation attack pretty unlikely since the separated parts of network may not have a single link for attack to be successful.

There are already mitigations in place for attacks with cancer nodes. In particular Bitcoin clients only make 1 outbound connection per 16-bit IP address network range[49]. This means that from 65 536 addresses from for example x.y.0.0 to x.y.255.255 only 1 is used by the client to connect to Bitcoin network. therefore an attacker wanting to flood the network with cancer nodes would have to have control over several machines with IP addresses in huge amount of different network ranges. This would be doable by an attacker having access to big botnet.

Another possible mitigation for this would be using trusted auditing nodes with static IP-s for clients to specifically connect to. Those nodes could connect to each other and keep block-chain up-to-date. They would also be able to detect if announced block-chain branches are built by attackers with a lot of computing power. This trust network within Bitcoin network would however go against protocol and the idea of not having to trust anybody in peer-to-peer financial system. It is also possible that the honest trust-nodes get compromised and this could create a mess. Having knowledge of a few geographically distributed locatable and constantly running honest nodes that can handle thousands of Bitcoin connections at the same time and possibility to specify connections to them as an optional networking feature in the Bitcoin client should however increase the security of the system and maintaining a database of such nodes with their IP-s listed in Bitcoin wiki could be taken into consideration.

3.4. Client-side attacks

As we have seen Bitcoin is difficult to attack as a system. As a financial system it is still a profitable target for successful hackers and therefore the attacks are directed towards the clients. Attacking the clients is possible since in a decentralized currency like Bitcoin users take more responsibility with getting the control over their finances. Since there is no centralized business to control Bitcoin securing users finances is also up to the users themselves. More control means more responsibility.

Client-side attacks include wallet theft, attacks on users anonymity, denial-of-service and client software exploits. We define clients as both end users as well as Bitcoin businesses like currency exchanges and briefly discuss more popular methods how attackers are able to steal money or otherwise hinder the usage of Bitcoin.

3.4.1. Wallet theft

As discussed earlier Bitcoin wallet is a file held on users hard-drive. This file holds the keys needed to receive and more importantly for an attacker spend the Bitcoins held on the machine accessed. Getting hold of this file means getting hold of someones Bitcoin balances and control over their finances. This file can be accessed with breaching the physical security or otherwise making contact with a device holding the wallet but in most cases it is distant activity over the network and usage of malware that helps criminals in stealing Bitcoins.

First Bitcoin targeting malware was Infostealer.Coinbit, a Trojan horse that lures the users to execute it. Upon execution it looks for Bitcoin wallet in Windows machines and e-mails it to attacker through a server in Poland[54]. The attack was reported by Symantec during the Bitcoin bubble in June 2011[55] and the 25 000 Bitcoin heist mentioned in wallet chapter was probably executed with the help of this piece of malware. After fairly trivial Infostealer.Coinbit that targets Windows users other malicious programs were spotted by anti-virus companies like DevilRobber Trojan which targets Mac computers and spreads with pirated software downloaded from torrent sites such as Pirate Bay. This is far more complicated malware. It also steals wallet files but in addition it mines Bitcoins, collects system information like shell and browser history, collects usernames and passwords[56]. This means that in case of more complicated pieces of Bitcoin stealers encrypting wallet files may not save the infected users from getting robbed as the malware can also plant a key logger and get hold of encryption keys. It is also possible that more known Trojans like Zeus may start including Bitcoin stealing capabilities by default as Bitcoin gets more and more popular.

Users are now able to encrypt their private keys with standard Bitcoin client starting from version 0.4. This feature was added shortly after the 25 000 Bitcoins were stolen and users can opt in the usage of encrypting the wallet with Advanced Encryption Standard symmetric-key algorithm. If keys are encrypted users have to enter their passphrase when sending Bitcoins[57]. This does mitigate some of the simpler attacks as hackers have to brute-force the encryption passwords to get to private keys used for sending Bitcoins, but if the passphrase is trivial then this is not a big hurdle for a motivated attacker. Furthermore as discussed some malware can also get the passphrase used for encryption and therefore the encryption can offer a false sense of security to some extent and in case users lose their strong passwords they also lose their Bitcoins.

In general using Bitcoins is not that much different from using banking or e-wallet system for users in terms of client-side security: it is not safe to use unpatched insecure machines and compromised devices lead to loosing funds. This means that best practices for keeping ones security apply. Users should not open suspicious files, not browse shady websites, keep their software up-to-date and be somewhat paranoid using the computer that has Internet access, even more so when there are wallet files that hold keys that give access to large amount of Bitcoins.

Bitcoin protocol also supports transactions with multiple signatures. This means that it is possible to combine different private keys for authorizing a transaction and an output of previous transactions can not be spent in new dealings before the requirements in the scripts section of this output are met. In theory it is even possible to use a combination of keys so that key A or both B and C are used to spend coins that are sent to address that supports multi-signature security or even more difficult scheme with multiple keys[4]. This enhancement makes it possible to issue a transaction from a computer and then getting a notification on a smartphone to confirm the transaction for example, making the wallet a lot more secure. Developments for implementing this feature have already been started for standard client[58]. Multi-level authentication will mitigate the threat of falling a victim of a wallet theft but also makes Bitcoin a bit more difficult to use and like wallet encryption this feature has to be opted in by the users.

End-users are not the only ones holding wallets that are nice targets for attackers. Besides the mentioned biggest Bitcoin currency exchange Mt. Gox hack there have been other high-profile attacks on Bitcoin services, some of which have specifically targeted wallet files. Most notable Bitcoin business hit by hackers is Bitcoinica, an exchange that enables forex-like market actions with contracts on rate differences and a possibility to sell short the Bitcoins that users do not own by backing the deal with their US Dollars. Bitcoinica lost its wallet files twice during a 3 month period. First their wallet was stolen alongside 7 other Bitcoin wallets from Linux cloud provider Linode that had its customer support interface exploited and the stolen support credentials were used to compromise the accounts on Linode that ran Bitcoin clients to serve their customers[59]. Second time Bitcoinica was successfully attacked on their Rackspace virtual server and lost balances on their hot wallet used to automatically pay out requested withdrawals[60]. The service also lost the information about customers accounts and transaction history to the attacker as they were deleted with destruction of the server instances and there were no up-to-date backups created[61].

Hot wallet is the wallet kept on online server and used for automatic transactions. This means that encryption and other simple mitigations that are put to place to avoid loosing funds will not help in most cases since attackers having access to this wallet have most likely compromised the server and are able to work out the encryption scheme from the source files or network traffic.

To avoid getting hacked Bitcoin service providers should secure both their public web applications as well as servers and network. For servers it is smart to limit both their physical as well as virtual access to minimal amount of people, especially for those outside the company. This means that using cloud service and virtual hosting providers should be avoided, since the temptation for employees of such companies to get hold of big amount of Bitcoins with small possibility of getting punished might prove to be too big to resist. Bitcoin businesses have to realize that they deal with financial systems and therefore running constant security checks and using third party security auditing to keep an eye on their security is strongly suggested.

Security problems for Bitcoin services have several non-technical reasons. First of all Bitcoins are both interesting and valuable to hackers. Secondly Bitcoin theft is not criminalized. By international law and standards Bitcoins are not money and criminals feel strong sense of impunity. First time we could see criminals being prosecuted for stealing Bitcoins is probably still far away and legal systems have to adopt to new currency when it gets more popular. Thirdly getting into developing Bitcoin businesses is fairly easy: there are a lot of open source projects and code examples as well as helpful and intelligent community to get support from. In addition there is no licensing, laws and regulations one needs to follow in order to start accepting Bitcoins as means of exchange or offer financial services dealing with Bitcoin. Low barriers in getting into Bitcoin service provider may also mean that the quality of the software and level of security is not very high as developers may not have needed security background and requirements for including security-minded people and run audits is non-existent. In fact in peer-to-peer currency system there is no-one to enforce any rules.

Unfortunately security breaches of Bitcoin businesses and individuals victimized by theft bring bad publicity to Bitcoin as a system. Although the protocol itself is designed to be pretty secure the public image is portrayed as somewhat dangerous financial system. Knowledgeable Bitcoin users can mitigate the threats of falling victims to hackers and should carefully choose the services they trust with their funds. Good thing about Bitcoin is that ultimately it is a peer-to-peer currency and one does not need to trust any of the service providers as banks to participate in financial dealings. They can just run Bitcoin client in their local machine, maintain the transactions database and validate all the transactions by themselves automatically running the software.

3.4.2 Attacking anonymity

A lot of interest in Bitcoin comes from the perceived anonymity of Bitcoin transactions and the fact that one can send funds online with no restrictions while not revealing their real world identities. This is important for criminals like drug dealers but also for individuals who might be repressed by their governments or just people respecting their own privacy. Whatever the reasons for people wanting to stay anonymous they have to understand that Bitcoin is pseudo-anonymous. The perception of anonymity comes from the fact that there are no registrations or credentials to join Bitcoin network and issue transactions. Coins are linked to addresses that look like random strings. At the same time all transactions are publicly available in the block-chain and therefore it is possible to attack the anonymity of Bitcoin users. This can be used by law enforcement to find criminals using the currency but also by criminals to find and identify wealthy individuals holding large amounts of Bitcoins.

To get to link Bitcoins to an identity there has to be a mapping point. One or several transactions or addresses have to be linkable to real-world objects. This can happen when connecting an IP to transaction, on transportation of goods via a shipping address, forum signatures with Bitcoin addresses, registration to services sites and giving them an address or sending them funds, receiving funds from currency exchange sites that ask for personal documents or many other means. A combination of this information can be used to create a mapping and add notes to Bitcoin flow in transactions to reveal real people using the coins.

Fergal Reid and Martin Harrigan showed that using a graphical presentation of the network and adding publicly available information with links made from block-chain and open-source intelligence it is possible to associate many public keys together and link the information with data external to Bitcoin network[62]. The result of analysis to break anonymity is in practice a graph with points being addresses and links between them transactions. The addresses themselves can be further investigated if they somehow link to any individuals or services by information already obtained. If a party with certain power would conduct such mapping they could probably get the data for user information from currency exchange sites as well as other services and therefore build a more complete picture and possibly be able to even name stealers of coins if the hackers have not been careful enough to take steps to stay anonymous.

Although Bitcoin is not implemented to be very anonymous it is possible to stay a step ahead of attacks against the anonymity. Bitcoin users can use as many addresses as they please. If someone uses a different address for all transactions the mapping completeness will fail for an attacker and the holes in their informational chart may hinder the process of unmasking a Bitcoin trader. The users with strong need for anonymity would also want to use Tor or similar services hiding their network traffic and machines location. Extra careful attention should be taken by people wanting to stay anonymous on giving out any piece of information to any services related to Bitcoin. Once the linking-point has been made the real possibility of mapping identities to addresses emerges. For extra anonymity there are Bitcoin mixing services: they take the coins sent in by users and mix them with other users coins and then send back to different address of the user wanting to mix up the Bitcoin flow trail and increase personal anonymity[63].

Bitcoin traffic is also not encrypted[59]. System itself uses strong cryptography but data sent in peer-to-peer network is plaintext. This does not create opportunities for man-in-the-middle attacks as faking digital signatures of ECDSA is currently practically impossible, but emerges some extra security concerns nevertheless. In particular this may have an effect on anonymity of users.

Users of Bitcoin receive and relay new transactions they get from the network so there is constant Bitcoin traffic to and from the machine running Bitcoin client. First person to announce a transaction is the one sending the coins in this transaction. Other nodes catch the packet with transaction and then pass it on to to nodes connected to them. Some of those nodes mine for coins and they add the transaction hash to the merkle tree and if lucky enough have it included in a block that is then announced to the network and new blocks are built on top of it, all confirming the transaction and enhancing the trust that this transaction is no longer reversible by an attacker who has a lot of computing power.

First person to send information about a transaction therefore also reveals their Bitcoin addresses. This may be a great mapping point to connect real-world identities to Bitcoin traffic and addresses. For someone to be able to make this mapping they need to have 3 pieces of information: good overview of someones network and in particular Bitcoin traffic, traffic of nodes connected to the client and the personal information of person being investigated. While an attack on anonymity using this method may be run using the help of cancer nodes to get a good picture of transaction flow in the network connected to a particular client a better chance to reduce anonymity is by tapping the network to monitor the whole traffic passing a node or better yet several related nodes. Someone capable of running such an attack would probably want to include co-operation with an Internet Service Provider, who also knows the name and real-world location of the network owner.

Mapping a Bitcoin user to real-world identity proves difficult in case they are really concerned about their anonymity but with enough motivation, resources and connections it is possible. This is not a concern for average Bitcoiners however and Bitcoin design goal is not to be truly anonymous. Maybe one could even say that it is positive that there is a theoretical way for mapping criminals to transactions so that governments would not have to start forcing a system for dropping transaction traffic.

3.4.3. Denial-of-service and client software security

There are several ways to create conditions that result in denial-of-service for one or several targeted users but in some cases possibly for the whole Bitcoin network. Those are the theoretical but somewhat impractical examples mentioned in attacks with computing power and cancer nodes. Targeting a user to cut it off the Bitcoin network could also mean using a vulnerability in a Bitcoin client software. With finding flaws in open-source software it may be possible for an attacker to overflow the client to make it shut down or even worse, send it data that would result in nasty code execution situations that could reveal private keys if unencrypted.

Denial-of-service attacks to knock out a client software would mean sending the node running the client either big amount of information or specially crafted input that would not get handled properly. Attackers sending in too much data too quickly or illegitimate transaction messages would have their connection dropped. therefore Bitcoin client has some built-in denial-of-service prevention[59]. This mitigation can be bypassed by sending in data from several malicious nodes rapidly, but limitations to this include the connections per IP space limits mentioned in attacks with cancer nodes.

A better chance for an attacker to disconnect a node from the Bitcoin network would be finding a vulnerability in the client software. No piece of software that has some level of complexity is totally secure against attacks. The fact that Bitcoin is open-source project adds two different views to its security. First of all everybody can read the code and look for malicious input cases not properly handled or find other types of security holes. At the same time people reviewing the code can also report and fix the given problems.

In May 2012 a critical vulnerability in Bitcoin software was announced and tagged as CVE-2012-2459. This vulnerability let attackers isolate victim from the Bitcoin network and cause creation of block-chain forks[64]. The denial-of-service possibility would have affected almost all users running the default client and it was reported and fixed quietly in the background patching major Bitcoin mining pools and services software before making the found flaw and security hot-fix public.

Good case of responsible disclosure and quick attention and fix for issue shows maturity of Bitcoin project and capabilities of core developers involved. This does not mean however that client software is and always will be secure and invulnerable. As complexity is added to the software itself for supporting cases like multi-signature transactions or other new features also the attack surface increases. At the same time the attention to system and motivation for attackers due to increasing possibility of financial gain with successful exploit grows larger.

If attackers are able to knock the nodes offline Bitcoin software can be restarted and rejoined to the network and after patches for such vulnerabilities are released the network would continue running with only some amount of financial loss. The loss would be resulted for miners losing the chance to look for block solutions. With less mining competition and smaller difficulty attacker would get a chance to mine more blocks and collect the resulting fees while several miners are constantly offline and it would also make it easier for the attacker to run attacks with computing power and double-spend the coins. Being able to launch such a large scale denial-of-service attack an attacker would need 0-day vulnerabilities found in Bitcoin client as well as supporting infrastructure to send exploits constantly to nodes in peer-to-peer network. Some of the mitigation to this attack is offered by other pieces of client software connecting to Bitcoin network since it is very unlikely to find exploits to all clients available.

Denial-of-service conditions in client software are not the ones we should be concerned of the most since the damage done is rather small and temporary. There is however always a possibility that one day attacker finds a way to remotely execute code using a vulnerability in the software. A buffer overflow or similar anomaly may be maliciously exploited to install malware, send Bitcoins or steal keys. This would not be a flaw in Bitcoin protocol or system design but it could create nasty consequences for the whole network. If an attacker found a way to build a transaction message that would trigger something unexpected in client software this would propagate in the network possibly even affecting all the users.

Solution for this is of course similar to other open-source software projects approach: knowledgeable users should proof-read source code, write good testing methods to catch corner cases where unexpected inputs could create problems and implement fix where possible problems are located. There is no guarantee of security for open-source software and for Bitcoin an exploitable vulnerability has bigger effect than a lot of other systems due to a possibility of rapid propagations in peer-to-peer network and systems financial meaning.

3.5. Attacks summary

Hash collisions are highly unlikely and related threats are mitigated in client software. Generating key pairs where resulting addresses collide is infeasible as long as quantum computing or cryptanalysis do not make a great breakthrough for attackers possible. Theoretical attack scenario like using weaknesses in ECDSA to find someones private key from their public key fail because getting the public key requires preimage attacks on 2 different currently unbreakable hash functions. Preimage attacks or improving hashing algorithms affect mining and result in attacker being able to launch an attack with computing power with lesser resources, but they could not be used to steal coins or bring the system down on their own and weakening the hash functions does not pose a threat to block-chains integrity. Bitcoin cryptography is very strong and its significant weakening in the near future seams unlikely.

Being a system with strong overall security Bitcoin sees a lot of attacks targeting the client systems, end users and businesses alike. Attacking users anonymity by analyzing and mapping public transactions database with outside information is possible, although difficult in case the user is careful. This is not however a concern for Bitcoin security as anonymity in itself is not a design goal. Researching wallet theft we came to a conclusion that trading in Bitcoin requires clients to be extra careful with their systems security, giving limited trust to as small amount of parties as possible and multi-signature transactions that will be introduced in the future greatly add to client-side security. We are not safe from vulnerabilities discovered in Bitcoin software, but the overall state of software security is good and community has shown ability to deal with possible issues.

Main concern for Bitcoin as a system is still the attack with a lot of computing power. It is hard to launch, but doable and problem is that there are no mitigations in place. For a mitigation rating blocks and automatically adding checkpoints to block-chain was suggested and this should be researched further, testing various rating algorithms in both theory as well as testing networks. This mitigation would also greatly help against possible damage done with cancer nodes and reduce threats of wide scale denial-of-service attacks bringing Bitcoin down. Another proposal was adding functionality for forcing the Bitcoin software to connect to trust-nodes in the network to mitigate the chance of getting trapped between cancer nodes and being denied valid Bitcoin transactions from being sent or received or finding transactions they received being reversed after regaining full connection to the network.

4. Summary

We have described a peer-to-peer cryptographic currencies working principles and mostly security. We have shown that although the cryptography behind Bitcoin is currently not breakable the system can be attacked with a lot of computing power or cancer nodes. Those attacks are with very high difficulty however and in reality hackers go after Bitcoin clients to steal their wallets with malware. We also showed that Bitcoin is not designed to be anonymous but a user wishing to keep their identity private has the possibility to increase their chances of doing so.

For Bitcoin enhancements and extra mitigations we offer ideas for auditing nodes within the network who keep the clients from trusting branch of transactions database that might be generated by attackers. We offer a way to mitigate possible problems caused by attacks with a lot of computing power by rating blocks and adding checkpoints to block-chain. Future work would have to include further research into the mitigations and their implementation as well as mathematic proof on cryptographic security of Bitcoin.

For businesses or individuals wishing to trade in Bitcoins because of low fees and limitations, high control over ones finances and conveniency the systems security is currently very good. From technical point of view it can be safely used to send, receive and store great amounts of value. All users must note however that with increased power over their finances they also take greater responsibility and steps have to be taken to secure both personal and corporate systems before trading in Bitcoins. For bigger business-cases working with security-minded people and running regular security audits on your services is highly recommended.

References

[1] Thomas, S. What is Bitcoin? [WWW] http://www.weusecoins.com (16.04.2012)

[2] Bits and bob. The Economist blog. [WWW] http://www.economist.com/blogs/babbage/2011/06/virtual-currency (29.04.2012)

[3] Bitter to Better — How to Make Bitcoin a Better Currency / S. Barber, X. Boyen, E. Shi, E. Uzun [WWW] http://crypto.stanford.edu/~xb/fc12/bitcoin.pdf (11.03.2012)

[4] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. [WWW] http://bitcoin.org/bitcoin.pdf (11.03.2012)

[5] Technical background of Bitcoin adresses. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Technical_background_of_Bitcoin_addresses (14.04.2012)

[6] Address. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Address (14.04.2012)

[7] Bitcoin project. Sourceforge. [WWW] http://sourceforge.net/projects/bitcoin (16.04.2012)

[8] bitcoinj. Google Code. [WWW] http://code.google.com/p/bitcoinj/ (03.05.2012)

[9] Voorhees, E. Bitcoin — The Libertarian Introduction. [WWW] http://evoorhees.blogspot.com/2012/04/bitcoin-libertarian-introduction.html (29.04.2012)

[10] Proof of work. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Proof_of_work (14.04.2012)

[11] Bitcoinwatch. [WWW] http://bitcoinwatch.com/ (12.05.2012)

[12] Why Pooled Mining. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Why_pooled_mining (28.04.2012)

[13] Researcher Discovers Distrobuted Bitcoin Cracking Trojan Malware. Infosecurity Magazine. [WWW] http://www.infosecurity-magazine.com/view/20211/researcher-discovers-distributed-bitcoin-cracking-trojan-malware (16.04.2012)

[14] Controlled Money Supply. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Controlled_Currency_Supply (29.05.2012)

[15] Willis, N. Bitcoin: Virtual money created by CPU cycles. [WWW] http://lwn.net/Articles/414452 (11.03.2012)

[16] Grigg, I. Financial Cryptography in 7 Layers. [WWW] http://iang.org/papers/fc7.html (11.03.2012)

[17] Grigg, I. Is Bitcoin a Triple Entry System? [WWW] https://financialcryptography.com/mt/archives/001325.html (11.03.2012)

[18] Poulsen, K. New Malware Steals Your Bitcoin. [WWW] http://www.wired.com/threatlevel/2011/06/bitcoin-malware (15.04.2012)

[19] Bits and bob. The Economist blog. [WWW] http://www.economist.com/blogs/babbage/2011/06/virtual-currency (29.04.2012)

[20] Nakamoto, S. Bitcoin v0.1 released. [WWW] http://www.mail-archive.com/cryptography@metzdowd.com/msg10142.html (11.03.2012)

[21] Davis, J. The Crypto-currency. [WWW] http://cryptome.org/0005/bitcoin-who.pdf (11.03.2012)

[22] Bowbrick, S. Past currency. [WWW] http://www.guardian.co.uk/technology/2003/feb/25/comment.comment (11.03.2012)

[23] Dai, W. b-money. [WWW] http://www.weidai.com/bmoney.txt (09.04.2012)

[24] Back, A. Hashcash - A Denial of Service Counter-Measure. [WWW] http://www.hashcash.org/papers/hashcash.pdf (09.04.2011)

[25] Chen, A. The Underground Website Where You Can Buy Any Drug Imaginable. [WWW] http://gawker.com/5805928/the-underground-website-where-you-can-buy-any-drug-imaginable (15.04.2012)

[26] Wallace, B. The Rise and Fall of Bitcoin. [WWW] http://www.wired.com/magazine/2011/11/mf_bitcoin/all/1 (11.03.2012)

[27] Mick, J. Inside the MegaHack of Bitcoin. [WWW] http://www.dailytech.com/Inside+the+MegaHack+of+Bitcoin+the+Full+Story/article21942.htm (11.03.2012)

[28] Greeberg, A. Wikileaks Asks for Anonymous Bitcoin Donations. [WWW] http://www.forbes.com/sites/andygreenberg/2011/06/14/wikileaks-asks-for-anonymous-bitcoin-donations (27.04.2012)

[29] Bitcoin no. of transactions per day. Blockchain.info. [WWW] http://blockchain.info/charts/n-transactions (28.04.2012)

[30] Satoshi Dice. [WWW] http://www.satoshidice.com (05.05.2012)

[31] Bitcoin Days Destroyed. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Bitcoin_Days_Destroyed (01.05.2012)

[32] Kaminsky, D. Some Thoughts on Bitcoin. [WWW] http://www.slideshare.net/dakami/bitcoin-8776098 (05.05.2012)

[33] Stallings, W. (2010). Cryptography and Network Security. 5th ed. New York : Prentice Hall.

[34] Secure Hash Standards. US National Institute of Standards and Technology. [WWW]

http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf (28.05.2012)
[35] Block hashing algorithm. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Block_hashing_algorithm (06.05.2012)
[36] Schneier, B. Security Pitfalls in Cryptography. [WWW] http://www.schneier.com/essay-028.html (09.05.2012)
[37] Bellare, M., Goldwasser, S. Lecture Notes on Cryptography. [WWW] http://cseweb.ucsd.edu/~mihir/papers/gb.pdf (09.05.2012)
[38] Schneier, B. Cryptanalysis of SHA-1. [WWW] http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html (09.05.2012)
[39] Buldas, A., Laur, S. Do Broken Hash Functions Affect the Security of Time-Stamping Schemes? [WWW] http://www.cs.ut.ee/~swen/publications/articles/buldas-laur-2006.pdf (28.05.2012)
[40] On the Collision Resistance of RIPEMD-160 / F. Mendel, N. Pramstaller, C. Rechberger, V. Rijmen. [WWW] http://www.cosic.esat.kuleuven.be/publications/article-1355.pdf (12.05.2012)
[41] ECDSA. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/ECDSA (12.05.2012)
[42] Kanellos, M. Moore's Law to roll on for another decade. [WWW] http://news.cnet.com/2100-1001-984051.html (12.05.2012)
[43] Secp256k1. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Secp256k1 (12.05.2012)
[44] Pietrosanti, F. Not every elliptic curve is the same: trough on ECC security. [WWW] http://infosecurity.ch/20100926/not-every-elliptic-curve-is-the-same-trough-on-ecc-security/ (12.05.2012)
[45] Yang, E. Z. The Cryptography of Bitcoin. [WWW] http://blog.ezyang.com/2011/06/the-cryptography-of-bitcoin/ (13.05.2012)
[46] Sasaki, Y., Wang, L., Aoki, K. Preimage Attacks on 41-Step SHA-256 and 46-Step SHA-512. [WWW] http://eprint.iacr.org/2009/479.pdf (15.05.2012)
[47] Nakamoto, S. Bitcoin forum. [WWW] https://bitcointalk.org/index.php?topic=360.msg3520#msg3520 (15.05.2012)
[48] Yang, E. Z. The Cryptography of Bitcoin. [WWW] http://blog.ezyang.com/2011/06/the-cryptography-of-bitcoin/ (17.05.2012)
[49] Weaknesses. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Weaknesses#Attacker_has_a_lot_of_computing_power (17.05.2012)
[50] Top500 project. [WWW] http://www.top500.org/lists/2010/11/performance_development (18.05.2012)
[51] Gimenez, S. Can 51% attack be detected and dealt with? [WWW] http://bitcoin.stackexchange.com/a/1065 (18.05.2012)
[52] Laurie, B. Decentralised Currencies Are Probably Impossible. [WWW] http://www.links.org/files/decentralised-currencies.pdf (18.05.2012)
[53] Andersen, G. Bitcoin forum. [WWW] https://bitcointalk.org/index.php?topic=55394.msg661179#msg661179 (29.05.2012)
[54] Poulsen, K. New Malware Steals Your Bitcoin. [WWW] http://www.wired.com/threatlevel/2011/06/bitcoin-malware (15.04.2012)
[55] Doherty, S. All your Bitcoins are ours … [WWW] http://www.symantec.com/connect/blogs/all-your-bitcoins-are-ours (20.05. 2012)
[56] Rashid, F. Y. New Mac Malware Part Trojan, Data Stealer, Spyware, BitCoin Miner. [WWW] http://www.eweek.com/c/a/Security/New-Mac-Malware-Part-Trojan-Data-Stealer-Spyware-BitCoin-Miner-313602/ (20.05.2012)
[57] Thoughts on bitcoin wallet encryption. Github. [WWW] https://gist.github.com/2759512 (20.05.2012)
[58] BIP 10. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/BIP_0010 (22.05.2012)
[59] Roberts, P. Cloud Service Linode Hacked, Bitcoin Accounts Emptied. [WWW] http://threatpost.com/en_us/blogs/cloud-service-linode-hacked-bitcoin-accounts-emptied-030212 (20.05.2012)
[60] Dima, B. Exchange Site Bitcoinica Hacked, US$90,000 Stolen. [WWW] http://www.hotforsecurity.com/blog/exchange-site-bitcoinica-hacked-us90000-stolen-1719.html (22.05.2012)
[61] Gone With The Cloud - Bitcoinica Made No DB Backups. Bitcoin Money. [WWW] http://www.bitcoinmoney.com/post/23748723157/bitcoinica-no-database-backups (25.05.2012)
[62] Reid, F., Harrigan, M. An Analysis of Anonymity in the Bitcoin System. [WWW] http://arxiv.org/pdf/1107.4524v2.pdf (22.05.2012)

[63] Mixing service. Bitcoin wiki. [WWW] https://en.bitcoin.it/wiki/Mixing_service (22.05.2012)
[64] CVE-2012-2459: Critical Vulnerability (denial-of-service). Bitcoin.org. [WWW] http://bitcoin.org/dos (23.05.2012)
[65] Johnson, D., Menezes, A., Vanstone, S. The Elliptic Curve Digital Signature Algorithm (ECDSA). [WWW] http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf (1.06.2012)

Appendices

Appendix A

Python script to calculate minimum time it takes to find RIPEMD-160 collision for Bitcoin addresses with starting computing power at Bitcoin networks total power and increasing according to Moore's law.

```
hashes = 2 ** 80                   # average amount of hashes to try
                                   # before collision
seconds_in_month = 3600 * 24 * 30  # seconds every month
hashes_in_second = 12 * (10 ** 12) # initial computing speed
months = 0                         # months spent hashing
while hashes > 0:
    hashes = hashes - hashes_in_second * seconds_in_month
                         # amount of hashes to try decreases
    months = months + 1
    if months % 18 == 0:
        hashes_in_second = hashes_in_second * 2
                         # every 18 months the hashing power doubles

print months / 12.0
```

Output: 16.5833333333

Appendix B

Bitcoin client C++ source code. Function handling incoming transactions from main.cpp line 473. Sources available from https://github.com/bitcoin/bitcoin/blob/master/src/main.cpp. Function returning false to indicate transaction unacceptable if the hash of this function is already seen in the block-chain, the database of all transactions.

```
bool CTxMemPool::accept(CTxDB& txdb, CTransaction &tx, bool
fCheckInputs, bool* pfMissingInputs)
{
/---/
    // Do we already have it?
    uint256 hash = tx.GetHash();
    {
        LOCK(cs);
        if (mapTx.count(hash))
            return false;
    }
    if (fCheckInputs)
        if (txdb.ContainsTx(hash))
            return false;
/---/
}
```

Appendix C

Bitcoin client C++ source code. Function handling incoming blocks from db.cpp line 518. Sources available from https://github.com/bitcoin/bitcoin/blob/master/src/db.cpp. The function returns index of the block by the hash of incoming block. If the block is already in the database returning index of the block, otherwise adding the block to chain before returning its index.

```
CBlockIndex static * InsertBlockIndex(uint256 hash)
{
    if (hash == 0)
        return NULL;

    // Return existing
    map<uint256, CBlockIndex*>::iterator mi =
mapBlockIndex.find(hash);
    if (mi != mapBlockIndex.end())
        return (*mi).second;

    // Create new
    CBlockIndex* pindexNew = new CBlockIndex();
    if (!pindexNew)
        throw runtime_error("LoadBlockIndex() : new CBlockIndex
failed");
    mi = mapBlockIndex.insert(make_pair(hash, pindexNew)).first;
    pindexNew->phashBlock = &((*mi).first);

    return pindexNew;
}
```