JIBC

Journal of Internet Banking and Commerce

An open access Internet journal (http://www.icommercecentral.com)

Journal of Internet Banking and Commerce, December 2017, vol. 22, no. 3

BLOCKCHAIN: BITCOIN WALLET CRYPTOGRAPHY SECURITY, CHALLENGES AND COUNTERMEASURES

ER-RAJY LATIFA

Department of Computer Sciences, University Cadi Ayyad, Marrakesh, Morocco

Tel: 212655939899;

Email: **errajy.latifa@gmail.com**

EL KIRAM MY AHEMED

Department of Computer Sciences, University Cadi Ayyad, Marrakesh, Morocco

EL GHAZOUANI MOHAMED

Department of Computer Sciences, University Cadi Ayyad, Marrakesh, Morocco

ACHBAROU OMAR

Department of Computer Sciences, University Cadi Ayyad, Marrakesh, Morocco

Abstract

Bitcoin has experienced rapid growth in the transactions number and in their value since its appearance in 2008. Its success is mainly due to the innovative use of a peer-to-peer network to implement all aspects of the currency life cycle, from creation to transfer between users. Bitcoin offers cash transactions that are almost instant and non-refundable, while allowing truly global transactions processed at the same speed as local ones. It offers a public transactions history, which allows untrusted audibility, and introduces many new and innovative use cases such as smart property, micropayments, contracts and escrow transactions for disputes mediation. However, the same features that make Bitcoin attractive to its end users are also its main limitations. Its decentralized nature limits the number of transactions and the speed at which transactions can be carried out and confirmed. The problem with slow confirmations is combined with the semantics of the confirmations which are not definitive, requiring several confirmations and further delaying the transaction acceptance. In this paper, we described the operating principles of peer-to-peer cryptographic currencies and especially security of bitcoin system. Moreover, For Bitcoin enhancements and additional mitigations we provide ideas for node auditing users in the network in aim to keep clients from the trusted transaction branch database generated by the attackers.

Keywords: **Component; Security; Challenge; Bitcoins; Countermeasure; Blockchain**

INTRODUCTION

A blockchain is a technology for a new generation of transactional applications that, through a collective consensus mechanism coupled with the use of a large, decentralized and shared public account book, builds trust, accountability and transparency while streamlining business processes [1]. A blockchain is simply a database that contains all the exchanges history between its customers since its creation. Without intermediaries, bitcoin various users share this database between them, it is secure and distributed at the same time, this advantage allows each user to check the channel validity.

By the end of 2008, by using the digital currency bitcoin, the first blockchain was developed by an unknown person gives himself the pseudonym Satoshi Nakamoto [2]. It is an underlying architecture. If blockchain and bitcoin were built together, today many actors (companies, governments, etc.) put in the consideration the exploit of blockchain technology for cases other than digital money. For companies, the absence of an intermediary - intrinsic to the blockchain has several advantages: it eliminates certain costs and simplifies and accelerates the procedures (no "paper" treatments or manual operations). It strengthens security since only the contracting parties have access to the stored data. A significant advantage in the Big Data era where the least exchanged data can be captured and analyzed by third parties [3].

Despite its short history, the blockchain has experienced some setbacks. Such as the piracy of the investment fund "The DAO" in June 2016, the bitcoins theft on the "Bitfinex" and "Mt. Gox" exchange platforms following computer flaws, this recent technological environment is still nebulous [4].

The highest risk with Bitcoin apart their financial risks is the loss or theft of an account. When the user loses his 256-bit key, there is no way to recuperate the bitcoins in his account. For example, in 2013, a British man threw a hard drive containing the key to unlock his 7500 bitcoins, valued at $ 7.5 million at the time, which are lost forever [5]. Bitcoin online trading services also offer wallet management services: they store private keys for the user, just like a bank stores his money. This is a way to transfer the risk of the individual to a Bitcoin business [6]. But hacking a remote computer and stealing a value of 256 bits is generally less risky and easier for thieves than stealing a bank and running away with cash bags. For example, in July 2011, 51% of funds held by my Bitcoin wallet on behalf of its member had being lost, in February 2014, the Bitcoin Exchange Platform, Mt Gox, stripped of the equivalent of $ 350 million in assets of members [7]. A 2016 study even forecasts that, between 2009 and 2015, 33% of Bitcoin's trade was pirated [8].

Similarly, the user may lose the Ethereum account key [9]. Nevertheless, the Ethereum complexity introduces additional risks: if a software or mobile application is unstable, it is decentralized and its version can also be uncertain. Worse still, attackers can exploit Ethereum vulnerabilities and compromise a portion of network assets. For example, in June 2016, an unknown attacker exploited a defect in the design of the Decentralized Autonomous Organization (DAO), an Ethereum application that serves as a venture capital fund. The attacker succeeded in transferring $60 million of Ether from DAO investors to his own accounts [10].

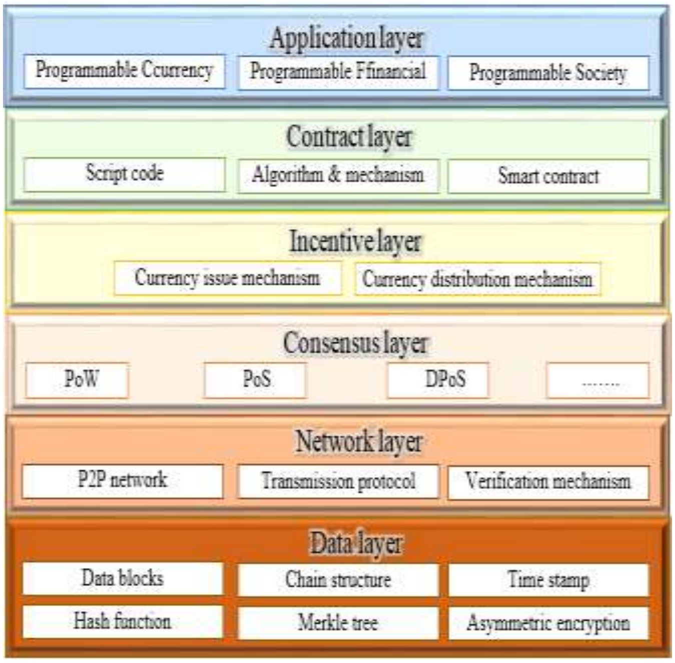
BACKGROUND

Blockchain Architecture

The Blockchain architecture is emergent decentralized and a paradigm of distributed computing that underlies Bitcoin and other cryptocurrencies. A blockchain is something like a general ledger in which all transactions have been recorded, and it is shared by participants in a bitcoin network [11].

Blockchain architecture (Figure 1) consists of a data layer, a network layer, a consensus layer, an incentive layer, a contract layer, and an application layer [12]. In addition, the data layer encapsulates the lower layer data blocks and the relevant asymmetric encryption and timestamp technologies. In the data layer, each node can use the hash function SHA, RSA, Merkle and so on? to encapsulate transactions and the code received in a certain time into a new block with timestamp, then, this new block is going connect to the main block in aim to be added a in the chain as a new block.

Figure 1: Blockchain architecture.



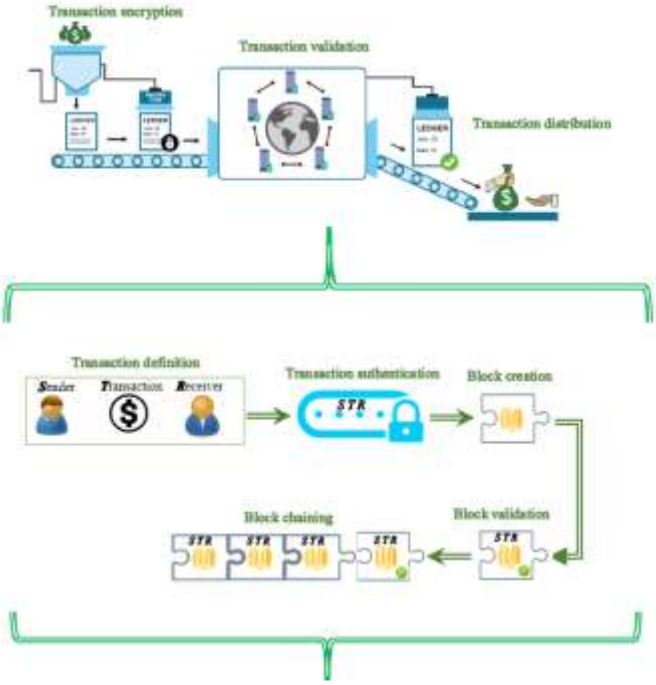
The network layer consists of a distributed network mechanism, a data transmission mechanism, and a data verification mechanism. The main role of network layer is allowing the participation of each node in the transaction registration and data verification process. If and only if a block is checked by most nodes of the network, it will be checked [13].

The consensus layer mainly encapsulates consensus algorithms for all nodes. It makes blockchain technology sensitive and reliable even in an efficient distributed network. The incentive layer combines economic factors within the blockchain, including the currency mechanism and the currency distribution mechanism to encourage the bitcoins miners (who generate the next block) [14]. Essentially, the incentive layer responds to the crowdsourcing problem between different distributed nodes. Therefore, there must be an effective crowdsourcing mechanism to ensure the maximum benefit of individuals, so that the safety of the entire chain-of-blocks system can be achieved. The contract layer network contains scripts, linked algorithms and intelligent contracts come from code and algorithms. The contract layer is the prerequisite for flexible programming and operation of data in a block chain system. The application layer embodies scenarios and application cases. With the support of the timestamp chain structure, distributed node consensus, PoW-based economic incentive and flexible programmable intelligent contract, the blockchain is technical and creative [15].

How it Works

Each transaction is composed of three mean elements: the sender, transaction information and receiver and it is guaranteed by an encryption code as shows Figure 2. The block contains several transactions and blockchain is built from multiple blocks [16].

Figure 2: How blockchain works.



The steps are:

Transaction definition: Is the first step, the sender creates a transaction containing the recipient's public-address information, the transaction value and a cryptographic digital signature that verifies the transaction validity and credibility.

Transaction authentication: In this step, the network nodes receive the transaction, then, they confirm the message by decrypting the digital signature. This message is held temporarily to be used in the block creation.

Building blocks: One of the nodes in the network updates the ledger or block using the pending transactions. Thus, within a specific time interval, the updated block is transmitted to the other nodes waiting for validation.

Block validation: once a request to validate an updated block is received by the nodes responsible for network validation, they use an iterative process that requires the approval of other nodes to authenticate the new block.

Block chaining: in this last step, the new state of the new block is transmitting to the rest of the network after that all the operations in a block are approved, and the new block is added "chained" to the blocks of the current chain.

Bitcoin Wallet

Bitcoin wallet is a file in users' file system. It holds public and corresponding private key pairs and transactions made from and to that wallet. The keys are used to receive and send Bitcoins. Public keys are given to payers to identify recipients and private keys are used to sign transaction messages and confirm currency exchange [17]. User preferences are also kept in wallet files that can and should be encrypted to mitigate the risk of parts loss to a hacker.

A Bitcoin address is a character identifier of 25-34 characters consisting of numbers and lowercase and lowercase letters. Most of the addresses used have 33 or 34 characters. The address usually starts with 1 and never contains the number 0 or the uppercase letter "O", or the lower case "l" or "I" for better legibility [18]. The address itself is removed from the public part of the Elliptic Curve Digital Signature Algorithms (ECDSA) key pair [19]. After several hashing cycles with the RIPEMD-160 and SHA-256 hashing algorithms, a checksum for the address is added and they are encoded with a modified Base 58 coding which results from the mentioned format.

An example of a Bitcoin address would be 1N3rjCLXhuuWFCweLV88GrDym4pryx7tkq. This can be easily validated and sending Bitcoins to an address that cannot have the corresponding private key and therefore cannot be used to receive the amount sent is quite unlikely. The probability that an incorrect address is accepted as valid is about 1 over 4.29 billion [20]. Therefore, there is good protection for typing errors, although typing addresses for sending Bitcoins is probably a rare opportunity.

The Bitcoin addresses and ECDSA key pairs both are not part of the structure of the Bitcoin network. They can be created safely offline after the hash and encoding rules described in the original design. The Bitcoin network will only know the address after its first use and a transaction has reported it. As the creation of addresses is simple and fast with several tools, including the original Bitcoin client [21], in order to improve these tools anonymity it is commonplace for users to exploit several thousands of addresses. Although sending Bitcoins to invalid addresses is not possible, a transaction can be made to an address where the wallet file with keys is lost due to negligent owner behavior, machine failures, or malicious activity. There is no way to use these parts again and they are lost forever. This is another reason why it is wise for Bitcoin users to keep secure backups of their wallet.

The original Bitcoin client is written in C++ and is open-source [22], but several other programs are available to connect to the Bitcoin network and participate such as Java client BitCoin J that download just the block headers [23]. By design, it is possible to ignore Bitcoin transactions since all the money received is already spent and therefore included in other transactions. These measures are necessary for Bitcoin to be scalable and to be used with high transaction volumes, comparable to the credit card transfer rates in the world today. One can also choose to use eWallet services [24] to avoid downloading the ever-growing blockchain with all transactions but these results in the elimination of some checks on their Bitcoins and eventually accepts higher transaction fees.

CRYPTOCAPHY STRONGHT IN BITCOIN SYSTEM

The strength of Bitcoins is that it uses cryptography in a way that no other system exists before it actually works. It is a currency that does not need a central part to manage it, everything is defined by the laws of mathematics. But, as Bruce Schneier says, the cryptographic system cannot be as strong as the algorithms on which it rests, and when one of them is broken; the system goes down [25]. This is especially true for Bitcoin since it is a highly built system on cryptographic knowledge. The failure of the algorithms for Bitcoin would mean that one of the main cryptographic systems was broken. These are ECDSA [26], SHA-256 [27] and RIPEMD-160 [28]. All are algorithms published with extensive research.

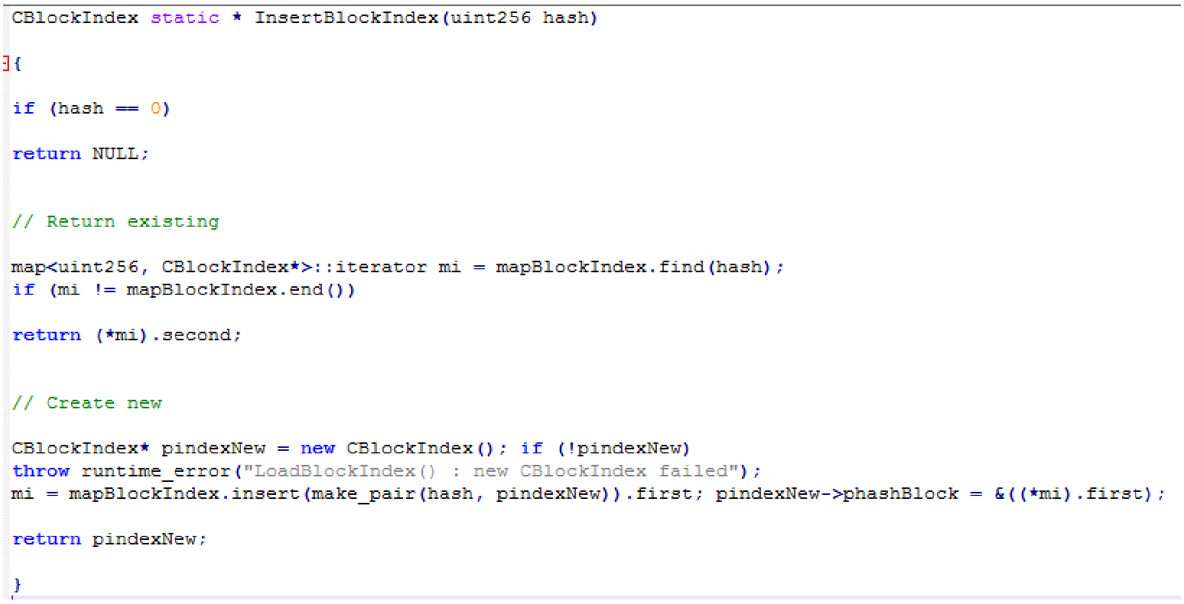
Collisions SHA-256

SHA-256 or other hash algorithms have two different attacks that we should be concerned about: collision and pre-attack. The collision is situation where different entries are chopped in the same synthesis value. Finding a collision for a SHA-256 via a raw force attack is possible because it has a limited amount of different hash values that it can produce. There are a total of 2256 results for hashing, so collisions are very unlikely to occur and we are not concerned with such a possibility. On average, a good attacker using the birthday paradox to his advantage is likely to find a collision in "only" 2128 tests for SHA-256 and we need much better to find a collision to consider a broken algorithm. If there is a simpler method for finding collisions than crude forcing because of the cryptanalysis, we consider that there is a defect in the algorithm [29].

In 2005, Chinese cryptographers burst SHA-1: they developed a method to find collisions 2000 times faster than the brute-forcing [30]. Their method has been surpassed by other cryptographic work and the machines have become much more powerful over the last 7 years, but finding a collision would still have a lot of computing resources and luck. If we think theoretically about a cryptographic system similar to Bitcoin but developed before 2005 and using SHA-1 as the main hash algorithm, which could lead to a breakdown of the function to the system 7 years after the first document was published how to find collisions faster than brutal forcing.

First of all, Bitcoin would not be theoretically sure if it was using SHA-1, but the attacks would still not be relevant to the practice and the search for holes that could be exploited in a system would not be easy. In Bitcoin hashing is mostly used in extraction and transactions [31]. For transactions, it is necessary to sign the transaction hash to transfer the value of the parts 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, that person can add himself as the receiver of the coins so then he is able to steal [32] (Figure 3).

Figure 3: Bitcoin client C++ source code, function handling incoming transactions from main.cpp line 473.



```
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;

}
```

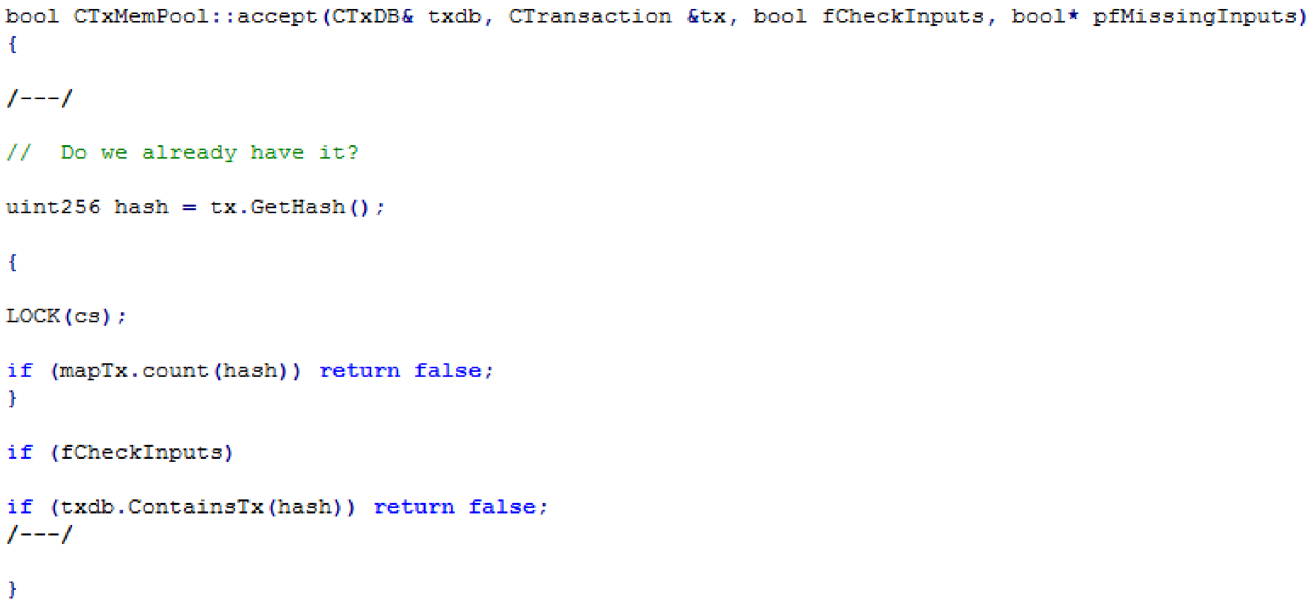
The original signature of the senders on the transaction hash would be valid and, therefore, the transaction would look like a valid transaction. The attacker should find this very specific collision instead of a simple collision: a transaction message that has its Bitcoin address instead of the destination address should have the same hash. Because these weaknesses found in SHA-1 are by far not enough [33]. In addition, the attacker must be faster than the owner of the coins by spending them.

Once the transaction is added to a block per minor, the attacker can use the outputs of the previous transaction entries and spend the coins. The two transaction messages refer to the same previous transaction with different scripts added with different owners having the right to spend the coins exploiting the previous results obtained in last transaction. If the attacker finds the transaction hash colliding after the original owner has expended the results, his efforts would become useless [32].

Since there is a remote possibility that several transactions may chop in the same summary in the future, attenuation is well developed before collisions in SHA-256 make this possible attack. The standard Bitcoin client does not add transactions to the database if he enters with a hash already saved. The transaction must be duplicated and deferred as described in Figure 3. For protocol in general, this could become a problem if some people start using thin clients that do not contain all transactions in their database.

The attacker would like to find collisions in block hashes to steal transaction fees and block the discovery bonus or invalidate some transactions or all transactions for denial-of-service or double spending attacks [34]. Unlike transaction blocks, they do not live alone. There may be two transactions with the same hash value in the block chain and both can refer to the same previous transaction in the event of a collision. For blocks, this is not possible because they form a time-stamped linear hash chain. The attacked blocks have the same hash value as one of the previous blocks would not be added to the chain if they were referring to the same previous block as the block they wanted to replace. Each new block has discovered previous blocks, hash and timestamp in its header and the blocks must be in chronological order. The standard Bitcoin client does not accept a block with a hash previously saved in the database. This code is shown in the Figure 4.

Figure 4: Bitcoin client C++ source code, function handling incoming blocks from db.cpp line 518.



```
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;
/---/

}
```

Buldas and Laur have shown that, to build a secure timestamping service, the hash functions used on the server side need not be resistant to collisions, resistant to preimages and not only one-way [35]. This means that, in terms of the integrity of the blockage breaking integrity, the hash algorithm of SHA-256 has no real effect. The old chain would remain unscathed from all hashed transactions where the hash algorithm change would be required for other reasons. Then the hash will continue with the last block resolved with the old hash as an input reference point again in the block that is approved by the community as the starting point of the new hash algorithm for the extraction process.

Another theoretically possible cryptographic attack is the improvement of the hash algorithm for SHA-256. If someone has found a way to find SHA-256 headers of larger blocks than others, he would gain an advantage in mining. He could get the monopoly to add blocks to the blockchain with the help of a large amount of computing power and reverse their own transactions or use it for denial of service against minors and regular users by building empty blocks and without including transactions. Improving the SHA-256 hash algorithm would have a possible effect on Bitcoin only if the improvements remain private. If many miners start using more efficient algorithms, difficulties extraction would increase and the system would continue to function normally.

Attack Transaction Signatures

We also have to look at any collisions in the RIPEMD-160. These are 296 times more likely to occur that collisions in SHA-256 as the length hash of 160 bits instead of 256 bits RIPEMD-160 is used to create Bitcoin addresses that are used to identify sent coins.

This means that if someone finds a pair of ECDSA keys where the public key is hashed to the same value RIPEMD-160 synthesis that other people Bitcoin, he could spend all the holding address pieces. But to create this type of collision, attacker would have to find a valid ECDSA key pair that would have hashed the RIPEMD-160 hash value in a collision and the hash process of a public key in an address involves first using SHA-256 and RIPEMD-160 before calculating a checksum with double SHA-256 and encoding a Bitcoin address [36].

In 2006, a team from the University of Technology in Graz showed that the methods used to find collisions in SHA-1 or RIPEMD did not extend against RIPEMD-160 and the algorithm was secure for known attacks [37]. This means that only the method of attack would be crude forcing that would generate key pairs ECDSA before chopping with SHA-256 and RIPEMD-160. So, in theory, using RIPEMD-160 makes the Bitcoin protocol less secure by offering shortening of public keys to be easily usable as addresses because of its shorter hash length, but in reality, the search process of collisions involves many calculations to make such an attack possible.

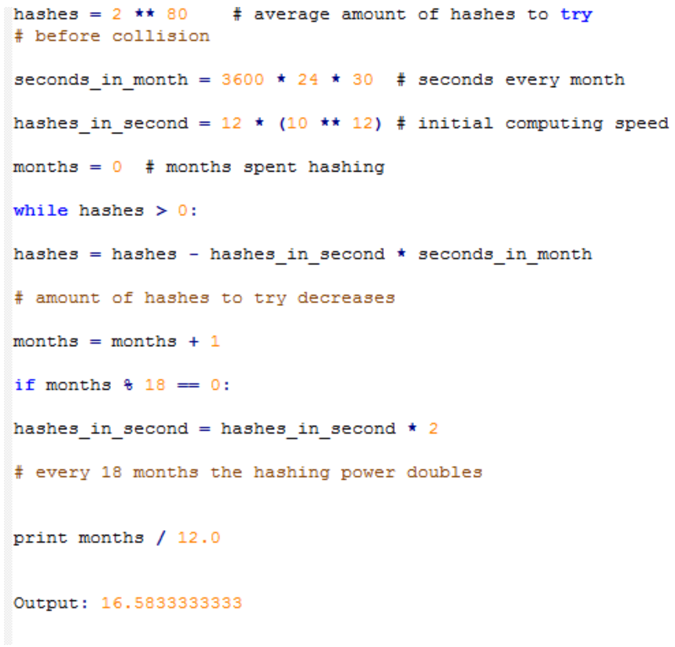
Then there are attacks against ECDSA. If someone could find a way to calculate private keys for key pairs where the corresponding Bitcoin address has funds sent to it, it could spend it as having the private key, signing transaction messages and transmit the value. The private key is an integer of 256 bits with 2256 different values and with this, it is more resistant to raw forcing than the Bitcoin address created from the public key because of hashing with RIPEMD-160 which has 2160 different values meaning this on average, Bitcoin address balances can be exchanged with 296 different key pairs [38]. The computational difficulty behind it is discussed in the previous paragraph: it would be necessary to calculate a public key for the private key, then to chop it twice with 2 different algorithms.

Let's look at the chances for an attacker trying to find the RIPEMD-160 hash value that runs into another Bitcoin address to be able to spend the pieces. On average, finding a collision would take a minimum of 280 hash tests. Let's say that an attacker has the same amount of computing power as all minors currently trying to solve a block that is about 12 tera hashes (12*1012 hashes) per second [39]. The total computing power of the Bitcoin system is calculated on the basis of mining production and in the case of mining, SHA-256 is calculated twice, as in most cases the use of the algorithm in Bitcoin takes the SHA-256 double from the input. Generously said that SHA-256 hashing in the extraction process takes the same time as any computer difficulty behind generating an address a private key would take for an attacker. We will see that on average, the attacker succeeds in 280/1012 seconds that is more than 38000 years. We will have to take into account that the computer power increases with time. We have doubled it every 18 months, as has often been cited in Moore's version of the law [40]. Now we will be able to find a private key in about 16.5 years, as shown in Figure 5. This is more than 16 years of constant hashing with very optimistic estimates to find a private key behind 1 particular Bitcoin address. As we see it today, crude forcing is impossible, but we need to monitor developments in cryptography, computer power and possibly even quantum computing, and be able to make adjustments in the algorithms used in the system.

If one tackles the ECDSA keys by brute-forcing is impossible, we must find a better method to attack the Bitcoins signature algorithm. Bitcoin uses the elliptic curve secp256k1 which has a 256-bit private key and is based on the Koblitz curve [41]. Algorithms using Koblitz curves are not part of the National Security Agency, ANSI or other standards and are therefore not widely studied and analyzed as some other ECDSAs. Therefore, it can be considered less secure and Fabio Pietrosanti suggests avoiding such an algorithm for these reasons [42]. Bitcoin seems to be the only widely used system that uses ECDSA based on the Koblitz curve and it seems that this is the part where the Bitcoin author may not have made the best choice, choosing speed on security. At the same time, no weaknesses are published for ECDSA and the keys are hidden behind the hashing algorithms. Suppose someone found a real weakness in ECDSA implemented in Bitcoin and was able to crack the algorithm and find private keys from public keys. Now attackers could forge signatures and therefore sign transaction messages with spare parts that they do not possess. But for Bitcoin attackers, it would not be possible to access these keys to steal money from users since they should first get the public keys to start calculating the private keys of these key pairs. However, public keys are hashed in the system. A successful preliminary attack on both RIPEMD-160 and SHA-256 is required before it is possible to use one of the weaknesses found in ECDSA, because public keys themselves are not broadcast in the network before the coins are signed at the next party and therefore spent [43]. Only Bitcoin addresses are available for attackers and these are constructed starting with the public key with SHA-256 and then with RIPEMD-160.

This means that only addresses that are reused are the subject of this attack because they revealed their public keys but this is not a problem because no weakness in ECDSA is known and users can increase their security and anonymity using different addresses for all transactions (Figure 5).

Figure 5: 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
```

Preimage Attack

The preimage attack on a hash function means to search for the original message from the hash value produced by the hash calculations [44]. In addition to the mandatory execution of the pre-imaginary attack to find private keys, Preimage would also help attackers reduce their coins faster. If they found a way to get rid of one of the hashish that encounters the difficulty required for a given block, they could present it as proof of work while collecting discovery fees and bonuses to find a new block and add it to the chain. This type of pre-imaginary attack would be interesting because there are several hashes that can be attacked, and the attacker can also control a part of the message that is going to be chopped. The attacker can change the Merkle root by deciding which transactions are added to the block and at what address the reward is sent. At the same time, he is only interested in finding the entire nonce value.

Currently, the best preimage attack for SHA-256 is against the 41-step version of the hash algorithm. The 64-step process is still secure against this meet-in-the-middle attack [45]. The attack of Meet-in-the-middle consists in attacking the hash function by working both ends of the hash at the same time. It tries to take the possible message values closer to the hash summary while taking the hash values closer to the original message until they are found in the middle and reveal the entrance of the hashing. In principle, this is exactly the type of attack that could succeed for Bitcoin because it is quite easy to find an appropriate nonce meet in the middle. An additional difficulty in launching a preimage attack is caused by the fact that the block headers use the SHA-256 double, but at the same time, a preimage that is found need not be specific: any help for hashing in one of the acceptable hash values is sufficient. This is one of the attacks that require more research, as there may be specific Bitcoin attacks in the possible medium against the SHA-256 double. If someone has found a method for this, it is highly likely that he would not publish it, because even a small advantage in mining is valuable.

At present, Bitcoins cryptography is very strong: crude forcing is infallible, algorithms are strong and, several mitigation measures which are already implemented in the event of weakening of the algorithms. With developments in cryptanalysis and calculation speeds, longer dimensions and hash lengths or better algorithms must be implemented in Bitcoin in the future. Although the creator of the system has announced the possibility of changing the cryptographic algorithms in the system in a transparent way for users in the unlikely event that SHA-256 breaks at any time, there is no concrete plan for do it [46].

Betcoin Walet Challenges

Betcoin nature: Nakamoto designed bitcoin system as a free source code to constantly introduce a steady supply of bitcoins to the market. To remove the bitcoins can switch from one account to another; the minor adds new bitcoins to the market by using special software to explore the Internet to search for bitcoin transactions that need verification [46]. This verification process involves solving complex mathematical problems requiring high levels of processing power. When the verification process end, a transaction fee of 25 bitcoins is collected by the minor, the checks frequency adds new parts to the market approximately every 10 minutes. Depending on the design, the verification process becomes more and more complex as more and more people inevitably try to exploit and invest in more powerful processors explicitly created to resolve mining problems. In addition, to be certain at least six blocks will have to be created in the blockchain, which makes the confirmation, time arises at one hour. Although this time is much less than the time required for merchants to receive payment from the customer via a credit card system, for customers, the situation is not similar. Making a client wait for an hour in order to make a payment is not realistic, especially when considering purchases directly at a retail store and not online. This disadvantage is one of the main reasons why other currencies have emerged crypto, such as Litecoin [47] for example. The time to find a block, and therefore be able to confirm a transaction, is much faster in these cases.

Although Bitcoin leaves the decision to set rights for a transaction to the user to make the payment (or the system/actor providing the payment system), since version 0.9.0 of the Bitcoin wallet, a minimal fee of 0.0001 BTC (0.1 CTMB) is part of transactions that have a size less than 1000 bytes [48]. This tax is still very low (around 0.04 euro), but can start to influence micropayments. Nevertheless, wallets may not choose to implement this option, but the transaction is not always certain to be approved.

In theory, as the number and sophistication of minor increase, the flow of bitcoins on the market can remain stable due to the difficulty of growing problem and a decreasing number of bitcoins assigned as the transaction fees. A constant flow avoids the exorbitant inflation which could otherwise occur with a growing demand and a limited offer. However, once all the 21 million bitcoins in existence are exploited, the coin flow on the market will cease, which is expected to occur around 2140 if the current rate of bitcoin extraction continues [49]. To provide some elasticity in the market supply of bitcoins, each piece can be subdivided up to eight decimal places. This ensures that as the demand for bitcoins increases, technology will be able to support a large user base. A key aspect of the bitcoin network extraction and transaction process is a permanent register of all transactions that have occurred since the bitcoin creation. Each time a minor check a transaction, its record is added to the blockchain, verifying that the involved bitcoins are not predisposed in a previous transaction. The minors are trying essentially to be the first to check block transactions and to add them to the blockchain, winning the reward Bitcoin [50].

This element of competition in the verification process ensures that countless miners look at each transaction, thereby unequivocally verifying that the bitcoins originate from an existing source and transfer to their designated destination. This certified system means that nobody can buy products with bitcoins that are not rightly and sit in their wallets.

In addition, the decentralized nature of the bitcoin system mitigates the risk of attack on the network itself because it is scattered on each computer that participates in mining. As Andreas M. Antonopoulos, a technology entrepreneur in the San Francisco Bay Area and one of Bitcoin's most open-minded supporters, said, "Bitcoin with no center means there is no of target to attack; there is no concentration of power. Power is diffused and distributed throughout the community [51].

Thousands of computers around the world are working together to update and maintain the blockchain, ensuring the accuracy and validity of every bitcoin transaction. Therefore, the bitcoin operation integrity will always be intact.

The transactions examined by the minors on the bitcoin network transfer directly from the consumer to the seller without intermediaries, transposing essentially a cash transaction on the Internet. Users are fully anonymous by this mechanism and third-party facilitators such as banks or credit companies become totally useless. However, this creates a risk as users must accept the responsibility of keeping their bitcoin stroboscopes in virtual wallets on a secure hard disk. Hackers can access a user's wallet if the computer's enclosure has an Internet connection, so that users have to be very careful about Internet security. Lack of central authority makes users more accountable for protecting their own assets. The vulnerability inherent in a decentralized system creates difficulties for bitcoin to reach a larger user base. To become a sustainable form, respected and widely used currency, Bitcoin will certainly need the government approval, in the form of a legal status. Risk aversion will prevent most people from investing their hard-earned dollars in a monetary system that operates outside the law limits and has no government guarantee to create inherent value. Unless the government provides a value guarantee, such as the legal status that gives to the dollar value, Bitcoin will always be perceived as an investment opportunity similar to a stock, which the price increases and decreases daily. However, cryptocurrency technology could prove revolutionary in the way monetary systems are exploited and transactions occur if governments decide to adopt the positive contributions that technology offers.

Attacks against Bitcoin System

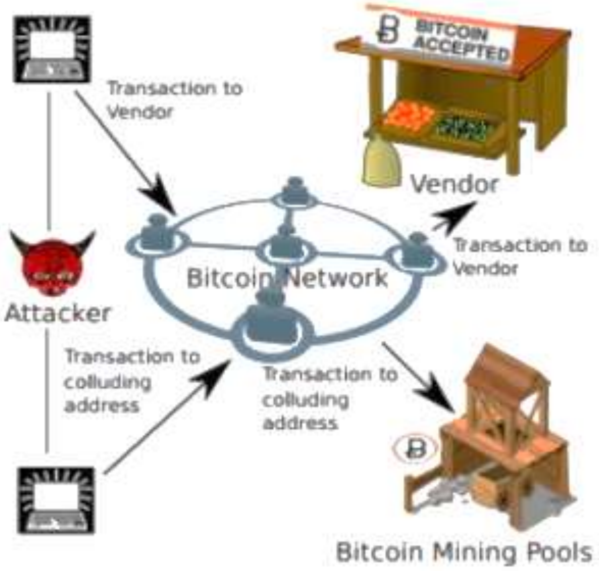
Attack with computer power: Bitcoin fights duplicate spending by adding all blockchain broadcast transactions. The blockchain is the database of all transactions and the branch of the chain with the highest computing cost is approved by the nodes in the peer-to-peer network [52]. Honest minors are based on the longest valid chain. They are rewarded by Bitcoins for doing this and in case they would suggest or, by chance, add blocks to the channel that is not considered a main branch through the network, the pieces they received by claiming the bonus block discovery and transaction costs would not be spendable since they are not included in the chain of trust. Clients should also trust only transactions included and confirmed by multiple blocks added to the chain after it. So, there is strong evidence that they are part of mainstream and not one of the orphaned chains that are not built on blocks that carry the greatest amount of calculations with them.

Double spending attack: Blockchain connection can occur in the event of an attack, but also by a chance when several new blocks are discovered and broadcast in a network at intervals of a few seconds. When this happens, the nodes in the network generating the blocks begin to build at the top of the block they received first. Now the block that is referenced by another new block will be part of the main chain and all the others will remain as orphans since there is more computational effort associated with this branch [53]. Transactions in orphan chains date back to the unconfirmed state and are added by miners building new blocks later.

The attacker who can produce a blockchain for which they show a proof of work, a level of difficulty corresponding to the chopping speeds and a greater amount of total computational effort than the constructors of the main chain would have control over the entire Bitcoin network. If an attacker is able to build such a string and broadcast the constructed chain, it would be accepted by the network as the main database of the transaction database. 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 to a block by a minor and consequently untrustworthy. Figure 6 demonstrations an example of double-spending attack.

As an effect of building a new main branch for the blockchain, attacker can reverse the transactions it signed that were added in the previous main branch to the point where the attacker split the chain [54]. The attacker does this by not simply adding transactions into a newly constructed branch and perhaps using the same pieces to issue other transactions, accordingly, spending them twice (Figure 6).

Figure 6: Example of double-spending attack.



The attacker could not reverse transactions that are not sent by him because he does not know the private keys with which it assigns the value to the other parties. He would not be able to create value out of the air, proof of work and rules of difficulty, building blocks must be followed even by creating an alternate block chain attack branch, and otherwise it is not accepted by other nodes. The attacker cannot take other people in cash because none of the transactions he adds to the created blocks that were not validly signed would be accepted for payment by other nodes on the network. In addition, these invalid transactions added to the block would also cause the block to be inadequate.

Denial of Service with Computing Power

What an attacker can do is not include transactions in his branch. These transactions would not have confirmed until they were added to the chain of blocks later. This could happen after the attacker loses most of the computing power in the network, stops attacking efforts, or begins to add transactions by other people in the transaction database created. Then the transactions would make it possible to entrust the level of confirmation required and the transaction would be valid unless someone could forge the chain with their computing power and create another branch after the previous division became the main chain and before the transaction is added to a block in the main chain [55]. This could result in a denial of service. Attackers can choose which transactions are added to the chain. They can in fact only add a redemption transaction in their blocks preventing any traffic that transmits value in the Bitcoin network rendering the system useless. If users can not send and receive payments, the currency is very unattractive. In this way, the attacker also loses transaction costs, but they may not be concerned that their goal is probably to kill Bitcoins in increasing popularity and if they keep control long enough, they can ultimately stop using money completely.

The attacker in control also prevents other minors from extracting valid blocks during the time they have most of the computing power while the other mining effort is branched out, which loses its main branch status in the blocks chain. The intelligent attacker would construct his chain in silence and would not broadcast the discovered blocks on the network. They should use more computer power than the combined Bitcoin network during this building in the background. Once they unexpectedly other users make their public efforts, their chain is accepted as the main chain by the Bitcoin protocol.

If this attack is carried out for long periods of time, attackers may lose total processing power if honest knots have overtaken it and it is unable to sustain itself. Then all his efforts will become useless and it is very unlikely that the community of Bitcoin ever knows that an attack has been launched. At the same time, the longer the control times of the attacker, the greater the damage of Bitcoin. Few hours of unconfirmed transactions would not create chaos, but over a week of reverse financial activity would allow average users to lose confidence in the system.

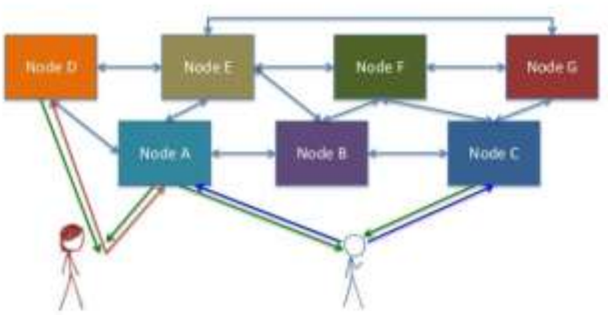
Cancerous Nodes

The attacking Bitcoin network or targeted users with cancer nodes would involve complementing the network with clients controlled by the attacker. The goal of this project would be to create a user or users to connect only to malicious nodes or to separate part of the Bitcoin network from others. Due to the network flooding with cancerous nodes, an attacker could refuse to relay blocks and transactions creating a denial of service. If he is also capable of segmenting the network, he can create a condition of several blockchains to be constructed simultaneously without knowledge of the others existence [56].

In case a successful network divides by running an enormous amount of cancer nodes, the attacker can double the coins in a manner similar to the methods discussed in the attacks with computing power with less effort. It will create a situation where a part of the network would grow on 1 branch and trust the transactions within that chain they think it is part of the main branch. In reality, after the cancer nodes disconnect and the network realizes that there has been a range in the blocks chain and solves it by choosing to trust the branch with the greatest amount of power total calculation put in place in the blocks within the limits specified by the protocol. Transactions in blocks now orphaned are not confirmed and, for some of them, the attacker may have been able to spend associated parts in another branch.

In the case where the network segmentation is not complete, the attack with the cancerous nodes fails. If the user that the attacker wants to disconnect from the network connects to an honest node that is in turn connected to the peer-to-peer network by at least one non-malicious node, it receives enough information about transactions and the blocks discovered to remain unharmed. This makes the total segmentation attack quite unlikely since the separate parts of the network may not have a single link for the attack to succeed. Figure 7 shows an example of network nodes used in bitcoin transaction between two users.

Figure 7: Example of network nodes used in Bitcoin transaction between two users.



There are already mitigation measures for attacks with cancerous nodes. In particular, Bitcoin clients make only 1 outbound connection per 16-bit IP address range [57]. This means that from 65,536 addresses, for example from x.y.0.0 to x.y.255.255, the client uses only 1 to connect to the Bitcoin network. Therefore, an attacker wanting to flood the network with cancer nodes should have control over several machines with IP addresses in a large amount of different network ranges. This could be done by an attacker with access to the big botnet.

Another possible mitigation for this would be to use trusted audit nodes with static IPs for clients that connect specifically. These nodes have the ability to connect to each other and keep the updated chain locked. They could also detect if the chains of announced blocks are constructed by attackers with much computational power. This trusted network within the Bitcoin network would go against the protocol and the idea of not having to trust anyone in the peer-to-peer financial system. It is also possible that honest trust nodes are compromised and this could create a mess. Having the knowledge of a few geographically distributed honest nodes that can handle thousands of Bitcoin connections at the same time and the ability to specify connections to them as an optional network feature in the Bitcoin client, this should increase security of the system and maintaining a database of these nodes with their IP listed in Bitcoin wiki could be taken into account [58].

The Client Software Security and Denial of Service Attack

To create the conditions that give rise to a denial of service, there are several ways to target individual users, but it is possible in some cases for the whole Bitcoin network. These are the theoretical but impractical examples mentioned in attacks with computer power and cancer nodes. Targeting a user to split it from the Bitcoin network could also mean using vulnerability in Bitcoin client software. By finding faults in open-source software, it is possible for an attacker to overflow the client to close it or even worse, send data that would result in malicious code execution situations that could reveal private keys if they are not encrypted.

Denial of service attacks to eliminate client software would mean sending the node that is running the client either a large amount of information or specially crafted inputs that would not be processed properly. Attackers who send too much data too quickly or illegitimate transaction messages have a link between them. Therefore, the Bitcoin client has an integrated prevention of denial of service [59]. This mitigation can be bypassed by sending data from multiple malicious nodes quickly, but limitations to this include the IP space limit connections mentioned in attacks with cancer nodes.

A better chance for an attacker to disconnect a node from the Bitcoin network would find vulnerability in the client software. No software that has a certain level of complexity is totally protected against attacks. The fact that Bitcoin is an open-source project adds two different views to its security. First of all, anyone can read the code and look for malicious cases of typing that are not handled properly or find other types of security holes. At the same time, people who review the code can also report and resolve problems given.

In May 2012, a critical vulnerability in the Bitcoin software was announced and marked as CVE-2012-2459. This vulnerability allows attackers to isolate the victim from the Bitcoin network and create block chain forks [60]. The possibility of denial of service would have affected almost all the users running the default client and it was reported and silently settled in the background by remedying the main Bitcoin mining and pool software before making public the best security fault and security.

A good case of responsible disclosure and a quick and corrective attention to the problem shows the maturity of the Bitcoin project and the capabilities of the main developers involved. This does not mean, however, that the client software is and will always be secure and invulnerable. As complexity is added to the software itself to support cases such as multi-signature transactions or other new features, that attack surface increases. At the same time, the attention paid to the system and the motivation for the aggressors due to the increasing possibility of financial gain with a successful exploitation increase.

If the attackers are able to topple offline nodes, the Bitcoin software can be restarted and joined to the network, and after the release of patches for these vulnerabilities, the network would continue to run with a single financial loss. The loss would result in miners losing the chance to look for block solutions. With less mining competition and less difficulty, attackers will have the opportunity to exploit more blocks and collect the resulting costs as several miners are constantly disconnected and would also facilitate the attacks with computer power and will spend the coins.

By being able to launch a large-scale denial of service attack, an attacker would need 0-day vulnerabilities found in the Bitcoin client and supporting the infrastructure to continuously send activities to the nodes in a peer-to-peer network. Some of the mitigation of this attack is offered by other software that connects to the Bitcoin network because it is unlikely to find exploits for all available clients.

Denial of service requirements in client software are not the ones we should be most concerned with, because the damage inflicted is rather modest and temporary. However, there is still a possibility that a one-day attacker can find a way to remotely execute code using vulnerability in the software.

A buffer overflow or similar anomaly can be exploited maliciously to install malicious software, send Bitcoins or flight keys. This would not be a defect in the Bitcoin protocol or system design, but this could create unpleasant consequences for the entire network. If an attacker has found a way to create a transaction message that would trigger something unexpected in the client software, it would spread across the network, which possibly will even affect all users.

The solution is obviously similar to that of other open source software: skilled users should read the source code, write good test methods to take cases where unexpected entries could create problems and implement possible problems if possible. There is no guarantee of security for open-source software and Bitcoin, a vulnerable exploit has a greater effect than many other systems because of the possibility of rapid network propagation and financial knowledge of the systems peer to peer.

Client-side Attacks

As we have seen, Bitcoin is difficult to attack as a system. As a financial system, it is still a cost-effective target for successful hackers and, as a result, attacks are directed at customers. Customer attack is possible because, in a decentralized currency, as users of Bitcoin take more responsibility to control their finances. Since there is no centralized company to control Bitcoin, securing users' finances depends on the users themselves. More control means more responsibility.

Client-side attacks [61] include worm theft, attacks on user anonymity, denial of service, and client software exploits. We define customers both as end users and by Bitcoin companies as currency exchanges and briefly discuss more popular methods on how attackers can steal money or harm the use of Bitcoin.

Wallet Theft

As mentioned earlier, the Bitcoin wallet is a file held on users' hard disks. This file holds the necessary keys to receive and, more importantly, for an attacker, spend the Bitcoins on the machine consulted. Obtaining this file means entering certain Bitcoin balances and controlling their finances. This file can be accessed in case of a physical security violation or to contact a device containing the walet, but in most cases, it is a remote activity on the network and use of software malicious people who help criminals to fly Bitcoins. The first BitCool targeting malware was Infostealer.Coinbit [62], a Trojan that attracts users to run it. When running, it searches the Bitcoin wallet in Windows machines and e-mails to the attacker via a server in Poland [63]. The attack was reported by Symantec at the Bitcoin bubble in June 2011 [64] and the 25,000 Bitcoin uprising was probably performed with the help of this malware. After a rather interesting Infotealer.Coinbit incident that targets Windows users, other malicious programs have been spotted by anti-virus companies like DevilRobber [65] Trojan that targets Mac computers and spreads with pirated software downloaded from torrents sites such as Pirate Bay. This is a much more complicated malware. It also destroys wallet files, but it also manages Bitcoins, collects system information such as shell and browser history, and collects user names and passwords. This means that in the case of more complicated pieces of bitcoin robots, encrypting wallet may not prevent infected users from being stolen because malware can also crash a key recorder and enter encryption keys. It is also possible that more well-known Trojan horses like Zeus can begin to include the default Bitcoin flight capabilities because Bitcoin is becoming more and more popular [66].

Users can now encrypt their private keys with the standard Bitcoin client as of version 0.4. This feature was added shortly after the 25,000 Bitcoins were stolen and users can opt for the use of wallet encryption with the Advanced Encryption Standard symmetric key algorithm. If the keys are encrypted, users must enter their passphrase when sending Bitcoins [67]. This mitigates some of the simpler attacks because hackers must brutally force encryption passwords to access private keys used to send Bitcoins, but if the passphrase is trite, it is not a big hurdle for a motivated attacker. In addition, as some malware can also get the passphrase used for encryption that can therefore provide a false sense of security to some extent and, when users lose their passwords, they lose their Bitcoins as well.

In general, the use of Bitcoins is not very different from the use of the banking system or electronic wallet for users in terms of client-side security: it is not safe to use unsecured machines and compromised devices that are not executed, resulting in the loss of funds. This means that best practices for keeping safety apply is that users should not open suspicious files, do not browse shaded websites, keep their software up-to-date and be somewhat paranoid using the computer that has Internet access, wallet containing keys that give access to a large quantity of Bitcoins.

The Bitcoin protocol also supports transactions with multiple signatures. This means that it is possible to combine different private keys to allow a transaction and a previous transaction output cannot be expended in new transactions before the requirements in the script section of that 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 parts that are sent to an address that supports multi-signature security or a more difficult with several keys [46]. This improvement makes it possible to issue a transaction from a computer and then to obtain a notification on a smartphone to confirm the transaction for example, which makes the wallet much more secure. Developments for the implementation of this functionality have already been launched for the standard client. Multi-level authentication will mitigate the threat of becoming a victim of a wallet theft, but makes Bitcoin a bit more difficult to use and, like wallet encryption, this functionality must be operated by users [68].

End users are not alone in having wallets that are good targets for attackers. Besides the largest currency exchange Bitcoin Mt. Gox [69] hack, other high-level attacks on Bitcoin services, some of which specifically targeted wallet files. The most notable Bitcoin Company hit by hackers is Bitcoinica [68], an exchange that allows market shares similar to forex with contracts on rate differences and an opportunity to sell Bitcoins that users do not own by supporting the agreement with their US dollars. Bitcoinica lost its wallet twice over a 3-month period. First, their wallet was stolen alongside 7 other Bitcoin wallets from Linode [70] who had operated the customer support interface and stolen media referrals were used to compromise accounts on Linode that had ran Bitcoin customers to serve their clients. Second time, Bitcoinica was successfully attacked on its virtual Rackspace server and lost balances on its hot wallet used to automatically pay the requested withdrawals. The service also lost information on client accounts and transaction history for the attacker since they were deleted with the destruction of server instances and no up-to-date backups were created.

Hot wallet is the purse used on the online server and used for automatic transactions. This means that encryption and other simple mitigation measures that are put in place to avoid losing funds will not in most cases occur because attackers with access to this wallet have likely compromised the server and can calculate the encryption scheme from source files or network traffic [71].

To avoid being pirated, Bitcoin service providers should secure their public Web applications, servers and network. For servers, it is wise to limit both their physical and virtual access to a minimum number of people, especially for people outside the company. This means that the use of cloud service and virtual hosting providers should be avoided because the temptation for employees of these companies to seize a large number of Bitcoins with a small possibility of being punished may prove to be too large to resist. Bitcoin companies must realize that they are dealing with financial systems and therefore performing constant security checks and using a third-party security check to monitor their security is strongly suggested.

Security issues for Bitcoin services have several non-technical reasons. First, Bitcoins is both interesting and valuable for hackers. Second, the theft of Bitcoin is not criminalized. According to international law and standards, Bitcoins are not money and criminals feel a strong sense of impunity. The first time we can see criminals being prosecuted for stealing Bitcoins is probably still far away and legal systems have to adopt a new currency when it becomes more popular. Third, entering into the business development of Bitcoin is quite simple: there are many open source projects and code examples as well as a useful and smart community to get support. In addition, there are no licenses, laws or regulations to begin accepting Bitcoins as a means of trading or offering financial services in Bitcoin. Low barriers in access to the Bitcoin service provider may also mean that the software quality and security level are not very high as developers may not need security information and requirements for including people with security problems and performing audits are non-existent. In fact, there is nobody there to apply the rules in the monetary system peer-to-peer.

Unfortunately, the security breaches of Bitcoin's companies and theft victims bring bad publicity to Bitcoin as a system. Although the protocol itself is designed to be fairly secure, the public image is represented as a dangerous financial system. Skilled users of Bitcoin can mitigate the threats of victims who fall against pirates and should carefully choose their services. Good thing about Bitcoin is that in the end it is a peer-to-peer currency and there is no need to trust any of the service providers as banks to participate to financial transactions. They can simply run the Bitcoin client in their local machine, keep the transaction database and validate all.

Attack Anonymity

A lot of interest for Bitcoin stems from the perceived anonymity of Bitcoin's transactions and the fact that everyone can send funds online without revealing their real identities to others. This is important for criminals such as drug traffickers, but also for people who could be repressed by their governments or simply people who respect their own privacy. Whatever the reasons why people wanting to remain anonymous, that they must understand that Bitcoin is pseudo-anonymous. The perception of anonymity stems from the fact that there are no records or credentials to join the Bitcoin network and issue transactions. Coins are linked to addresses that resemble random strings. At the same time, all transactions are accessible to the public in the chain of block and it is therefore possible to attack the anonymity of users Bitcoin. This can be used by law enforcement to find criminals using currency, but also by criminals to find and identify wealthy individuals holding large amounts of Bitcoins.

To be able to link Bitcoins to an identity, there must be a mapping point. One or more transactions or addresses must be linked to real world objects. This can occur when connecting an IP address to a transaction, shipping goods through a delivery address, signing up forums with Bitcoin addresses, registering on service sites, give them an address or send money, receive funds from exchange sites that ask for personal documents or many other means. A combination of this information can be used to create a map and add notes to the Bitcoin flow in transactions to reveal real people using the coins [72].

Reid and Harrigan have shown that using a graphical layout of the network and adding publicly available information with links created from block and open source information can combine multiple public and to link information with external data to the Bitcoin network [73]. The result of the analysis to cancel the anonymity is in practice a graph with address points and links between these transactions. The addresses themselves can be studied further if they relate in any way to individuals or services by information already obtained. If a party with some power would do such mapping, they could probably get the data for user information from the currency exchange sites as well as other services and thus build a more complete picture and maybe even naming the parts robots if the hackers had not been careful enough to take steps to remain anonymous.

Bitcoin traffic is also not encrypted [73,74]. The system itself uses strong cryptography, but the data sent in the peer-to-peer network is in plain text. This does not create opportunities for human attacks in the environment because the false digital signatures of ECDSA are currently practically impossible, but nevertheless emerge some additional security concerns. In particular, this may have an effect on the anonymity of users.

Bitcoin users receive and relay new transactions they get from the network, so that there is constant Bitcoin traffic to and from the machine that is running the Bitcoin client. The first person to announce a transaction is the one who sends the coins in this transaction. Other nodes capture the packet with the transaction, and then pass the nodes connected to them. Some of these nodes deserve coins and they add the transaction hash to the merkle tree and, if lucky enough, it is included in a block that is then advertised to the network and that new blocks are built. In addition, all confirming the transaction and improving the confidence that this transaction is no longer reversible by an attacker who has a lot of computer power.

The first person to send information about a transaction also reveals its Bitcoin addresses. This can be an excellent mapping point for connecting real world identities to Bitcoin traffic and addresses. In order for someone to do this mapping, they must have 3 pieces of information: a good overview of the network of some, and in particular the Bitcoin traffic, the traffic of the nodes connected to the client and the personal information of the person studied. While an attack against anonymity using this method can be executed using cancer nodes to obtain a good picture of the flow of transactions in the network connected to a particular client, a better chance of reducing anonymity is to use the network to monitor all traffic one node or better yet several related nodes. Someone in such an attack would probably want to include cooperation with an ISP (Internet service provider), who also knows the actual name and location of the network owner.

The mapping of a Bitcoin user to the real-world identity is difficult in the event that they are really concerned about their anonymity but with sufficient motivation, resources and connections. This is not a concern for average bitcoiners, but the design goal of Bitcoin is not really anonymous. Perhaps one could even say that it is positive that there is a theoretical way to map criminals to transactions so that governments should not start to force a system of declining transaction traffic.

CONCLUSION

In this article, we have described the operating principles of peer-to-peer cryptographic currencies and especially security and we have shown that although the cryptography behind Bitcoin is not currently broken, the system can be attacked with a lot of computer power or cancer nodes. These attacks are very difficult and, in reality, hackers go after Bitcoin customers to steal their wallets with malicious software. We also showed that Bitcoin is not designed to be anonymous, but a user who wants to keep his / her private identity can simply do it.

For Bitcoin enhancements and additional mitigations, we provide ideas for node auditing users in the network not to keep clients from the trusted transaction branch database generated by the attackers. We offer a way to alleviate possible problems caused by attacks with a lot of computing power by logging blocks and adding checkpoints to the blocks chain. Future work will be to include further research on mitigation measures and their implementation, as well as a mathematical proof of Bitcoin's cryptographic security.

REFERENCES

1. Lewenberg Y, Sompolinsky Y, Zohar A (2015) Inclusive block chain protocols. Int Conf Financ Cryptogr Data Secur, pp: 528-547.
2. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system, pp: 1-9.
3. Walch A, Bayern S, Christopher CM, Grinberg R, Marks C, et al. (2015) The bitcoin blockchain as financial market infrastructure : a consideration of

operational risk, pp: 837-894.

4. Andersson G, Wegdell A (2014) Prospects of Bitcoin. Spring Lund Univ Sch Econ Manag Dep Econ.
5. Shankland S. UK man tries to retrieve $7.5 million in bitcoins from dump James Howells' digital currency was worth almost nothing when he got it in 2009. Now his 7,500 bitcoins are on a hard drive buried in a landfill.
6. Hara K (2015) Upon emptiness. Digit Citiz.
7. Aumasson JP, Jovanovic P (2016) Blockchains in 2016 : status quo and scaling challenges. Kudelski Secur Decentralized Distrib Syst lab, EPFL, pp: 1-6.
8. Tandel S (2017) Blockchain: overview, use cases and challenges.
9. Wood G (2013) Ethereum: a secure decentralised generalised transaction ledger.
10. Del Castillo M. The DAO attacked: code issue leads to $60 Million ether theft.
11. Narayanan A, Bonneau J, Felten E, Miller A, Goldfeder S (2016) Bitcoin and Cryptocurrency Technologies. Princeton University Press.
12. Tuan T, Dinh A, Wang J, Chen G, Liu R, et al. (2017) Blockbench: a framework for analyzing private.
13. Li M, Weng J, Yang A, Lu W, Zhang Y, et al. (2015) CrowdBC: a blockchain-based decentralized framework for crowdsourcing, pp: 1-14.
14. Romano D, Schmid G (2017) Beyond bitcoin: a critical look at blockchain-based systems, pp: 1-31.
15. American Bar Association Criminal Justice Section and Center for Professional Development (2017) Blockchain Technology and Digital Currency National Institute.
16. Nian LP, Lee D, Chuen K (2015) Introduction to bitcoin. Handb Digit Curr, pp: 5-30.
17. Eskandari S, Barrera D, Stobert E, Clark J (2015) A first look at the usability of bitcoin key management full version.
18. Ron D, Shamir A (2012) Quantitative analysis of the full bitcoin transaction graph.
19. Bos JW, Halderman JA, Heninger N, Moore J, Naehrig M, et al. (2014) Elliptic curve cryptography in practice.
20. Androulaki E, Karame GO, Roeschlin M, Scherer T (2013) Evaluating user privacy in bitcoin.
21. Decker C, Wattenhofer R (2013) Information propagation in the bitcoin network.
22. Neudecker T, Andelfinger P, Hartenstein H (2015) A simulation model for analysis of attacks on the bitcoin peer-to-peer network.
23. Bartoletti M, Bracciali A, Lande S (2017) A general framework for bitcoin analytics.
24. Mjolsnes SF, Rong C (2003) On-line e-wallet system with decentralized credential keepers. Mob Networks Appl, pp: 87-99.
25. Grance T, Kuhn R, Landau S (2004) Cryptographic hash standards, pp: 2004-2007.

26. Johnson D, Menezes A (2000) The elliptic curve digital signature algorithm (ECDSA).
27. Gilbert H, Handschuh H (2004) Security analysis of SHA-256 and sisters, pp: 175-193.
28. Chiu-Wah Ng KWY (2004) A unified architecture of MD5 and RIPEMD-160 hash algorithms. IEEE Int Symp Cirquits Syst Vancouver, BC, 2: 889-892.
29. Goldwasserz MBOGS (1995) Incremental cryptography: the case of hashing and signing. Adv Cryptol, Springer-Verlag Y Desmedt 839: 1-19.
30. Schneier B (2005) Cryptanalysis of sha-1.
31. Zohar BYA (2013) Bitcoin: under the hood.
32. Rasmussen KB (2016) On Bitcoin security in the presence of broken crypto primitives.
33. Hoffman P, Schneier B (2005) Attacks on cryptographic hashes in internet protocols. Counterpane Internet Secur, pp: 1-12.
34. Vasek M, Thornton M, Moore T (2014) Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. Int Conf Financ Cryptogr Data Secur, pp: 57-71.
35. Buldas A, Laur S (2006) Do broken hash functions affect the security of time-stamping schemes? Proceeding 4th Int Conf ACNS 2006, Singapore.
36. Evans-Pughe C, Novikov A, Vitaliev V (2014) The bankruptcy of bitcoin's famous Mt. Gox exchange has raised doubts about the cryptocurrency's future, we look at the latest developments in the world of cryptocurrencies and visit the world's first Bitcoin Embassy in Montreal. IT Cryptocurrency, pp: 82-85.
37. Mendel F, Pramstaller N, Rechberger C, Rijmen V (2006) On the collision resistance of RIPEMD-160. Inf Secur Springer, pp: 101-117.
38. Goldfeder S, Bonneau J, Kroll JA, Kalodner H, Felten EW (2011) Securing bitcoin wallets via a new DSA/ECDSA threshold signature scheme.
39. Grinberg R (2011) Bitcoin: an innovative alternative digital currency.
40. Moore G (2015) Moore's law. Int Technol.
41. Bjoernsen K (2009) Koblitz curves and its practical uses in bitcoin security, pp: 1-4.
42. Pietrosanti F (2011) Not every elliptic curve is the same: trough on ECC security.
43. Ruffing T, Moreno-sanchez P, Kate A (2014) CoinShuffle: practical decentralized coin mixing for bitcoin. Eur Symp Res Comput Secur, pp: 345-364.
44. Clark J, Essex A (2012) CommitCoin: carbon dating commitments with bitcoin. 16th Int Conf, Kralendijk, Bonaire, pp: 1-8.
45. Biryukov A (2011) Meet-in-the-middle attack. Encycl Cryptogr Secur, pp: 772-773.
46. Nakamoto S (2013) Bitcoin a peer-to-peer electronic cash system.
47. Sunnarborg A (2016) Fundamental asset overview. Lawnmower Mark Res.
48. Kaskaloglu K (2014) Near zero bitcoin transaction fees cannot last forever. Soc Digit Inf Wirel Commun, pp. 91-99.

49. Faggart E (2015) What happens to bitcoin miners when all coins are mined? https://news.bitcoin.com/what-happens-bitcoin-miners-all-coins-mined/
50. Wang L, Liu Y (2015) Exploring miner evolution in bitcoin network. Int Conf Passiv Act Netw Meas, pp: 290-302.
51. Hill A (2014) Bitcoin: is cryptocurrency viable? Claremont Coll.
52. Courtois NT, Bahack L (2013) On subversive miner strategies and block withholding attack in bitcoin digital currency.
53. Karame GO, Androulaki E (2011) Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. ACM Conf Comput Commun Secur, pp: 906-917.
54. Rosenfeld M (2014) Analysis of hashrate-based double-spending, pp: 1-13.
55. Johnson B, Laszka A, Grossklags J, Vasek M, Moore T (2014) Game-theoretic analysis of ddos attacks against bitcoin mining pools. Int Conf Financ Cryptogr Data Secur, pp: 72-86.
56. Grayscale L (2016) Investments, bitcoin investment trust 2016 annual report. Shares Represent Common Units Fract Undivided Benef Interes.
57. Wang L, Pustogarov I, Tech C (2017) Towards better understanding of bitcoin unreachable peers.
58. Bitcoin wiki (2014) Satoshi client node discovery. https://en.bitcoin.it/wiki/Satoshi_Client_Node_Discovery
59. Vasek M, Bonneau J, Castellucci R, Keith C, Moore T (2016) The bitcoin brain drain : examining the use and abuse of bitcoin brain wallets.
60. Younis A, Malaiya YK, Ray I (2015) Assessing vulnerability exploitability risk using software properties. Softw Qual J.
61. Vyas CA, Lunagaria M (2014) Security concerns and issues for bitcoin. Int J Comput Appl, Natl Conf cum Work Bioinforma Comput Biol NCWBCB, pp: 10-12.
62. Plohmann D, Gerhards-Padilla E (2012) Case study of the miner botnet. 4th Int Conf Cyber Confl C Czosseck.
63. Panagiotis D (2015) Memory forensics and bitcoin mining malware: expanding the volatility framework for recovering bitcoin keys and addresses from ram acquired from multiple operating systems. Inf Commun Syst.
64. Coogan P (2011) Bitcoin botnet mining. https://www.symantec.com/connect/blogs/bitcoin-botnet-mining
65. Sybil B (2016) An introduction to devilrobber trojan.
66. Wikipedia (2009) Bitcoin. Free Encycl.
67. Gao X, Clark GD, Lindqvist J (2015) Of two minds, multiple addresses, and one history : characterizing opinions, knowledge, and perceptions of bitcoin. Across Groups, pp: 1-21.
68. Eskandari S, Barrera D, Stobert E, Clark J (2015) On the usability of bitcoin key management.
69. Kobayashi N (2015) Notice of commencement of filing of bankruptcy claims by MTGOX bitcoin exchange users. MTGOX Co. Ltd., pp: 1-10.
70. Slattery T (2014) Taking a bit out of crime: bitcoin and cross-border tax evasion, Brooklyn J Int Law.

71. Chirgwin R (2016) Linode: back at last after ten days of hell geo-blocks half the world to stop the DoS. https://www.theregister.co.uk/2016/01/04/linode_back_at_last_after_ten_days_of_hell/
72. Goldfeder S, Felten EW, Kroll JA. Securing bitcoin wallets via threshold signatures.
73. Reid F, Harrigan M (2013) An analysis of anonymity in the bitcoin system. Secur Priv Soc Networks, pp: 197-203.
74. Narayanan A, Miller A, Clark J, Kroll JA, Felten EW (2014) Mixcoin: anonymity for bitcoin with accountable mixes. Int Conf Financ Cryptogr Data Secur, pp: 486-504.