The Cryptography Behind Bitcoin

A look into the intricacies and the reliability

of the security behind the most popular

cryptocurrency coin to date

By

Alan Mazurek

Senior Thesis

SUNY Purchase

10/22/2020

Advisor: Professor Abdul-Quader

Second Reader: Professor Tunyan

Email Address: Alan.Mazurek@purchase.edu

Table of Contents

1. ABSTRACT …….. 3
2. INTRODUCTION ……. 3
3. BLOCKCHAIN ……. 4
   3.1 Proof of Work ……. 6
   3.2 Data Tampering …….. 7
4. ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM ……. 8
   4.1 Elliptic Curve Cryptography ……... 9
5. BITCOIN ADDRESS …… 10
   5.1 Generating Bitcoin Wallets …… 11
6. VULNERABILITIES …….. 12
   6.1 Double Spending …… 13
   6.2 51% Attack ……. 13
   6.3 Timejacking Attack …… 14
   6.4 Spam Transactions ……. 14
7. DISCUSSION …….. 15
8. CONCLUSION …… 16
9. REFERENCES ……. 16

1 ABSTRACT

With the growing popularity of cryptocurrencies around the world, Bitcoin's popularity and value continues to stand out among the rest. With its release dating back to 2009, the identity of Bitcoin's creator is still unclear to this day. Since then, cryptocurrencies have become even more accessible which in turn, has led to the acceptance of Bitcoin as a form of payment for several markets. It is estimated that there are about 100 million Bitcoin "users" around the world. This research will serve as a guide to explore the intricacies behind Bitcoin's uprising and its building blocks. Understanding the overall idea behind cryptocurrencies is crucial in learning how Bitcoin functions. Once we identify the technology behind Bitcoin, we will examine some of the important algorithms that play a crucial part in Bitcoin's security. We will also acknowledge how this cryptocurrency makes and tracks transactions at every step of the way. With the sudden surge in popularity, it is necessary to also address the vulnerabilities. This research will also consist of some theoretical cases that will help clearly outline why attacks/hijacks of Bitcoin possessions are rare and possible but are very expensive to pull off. This research will help bring a clear view and judgement to the overall reliability of Bitcoin and other cryptocurrencies.

2 INTRODUCTION

As more technological advancements are being discovered, the more the reason for certain aspects of our society to shift their infrastructure online. Over recent years, many people of society have become more dependent on the internet for personal and work-related usage. In 2008, Satoshi Nakamoto released a white paper outlining a new project he had been working on.

His vision consisted of creating an online form of payment that could be sent directly from person to person without needing to go through a financial institution (Oluwoye 22). He figured this would serve as a more efficient and easily accessible form of payment compared to the traditional cash and coin alternative. This alternative came to be known as Bitcoin, the first successful cryptocurrency. Nakamoto had finally developed a system as a free source code that was able to continuously pump out a steady supply of Bitcoin to the market (Latifa 13). This new form of paying online otherwise known as cryptocurrency, caught many peoples' attention. As more people began to rely on this new form of payment, the security behind this cryptocurrency coin became a hot topic. How could this new form of payment be secure if the Bitcoin wallets are public information?

3 BLOCKCHAIN

Before we explore the security that underlies Bitcoin, we must first address what Bitcoin means to cryptocurrency. Every type of cryptocurrency is created by a system known as blockchain. It can be simply thought of as a database that contains all the transaction history between customers since the creation of said cryptocurrency (Latifa 2). There exists a public document in this database known as a ledger, that records transactions that are made with Bitcoin. For every transaction on this ledger, it contains the following information:

1) The amount of Bitcoin that was sent or received
2) How much Bitcoin the original sender/receiver has in total *Fee not included*
3) The cost or fee for this specific transaction

These ledgers need to be sent to the Bitcoin network to validate the transaction and keep them as a record. The Bitcoin peer-to-peer network in its simplest form, is a gateway between two

personal computers that share resources and information with one another with no central authority/server present (Oluwoye 7). With a connection already established, one of the computers begins working on an incoming ledger. It is crucial to note that there exists a separate ledger which serves as a reward for the computer who takes on and completes a verification process required on the ledger. One of these ledgers is put to the side as a reward for whoever completes the work needed in this group. When a ledger initially reaches the Bitcoin network, a process known as "Proof of Work" needs to be done in order to validate the ledgers. This essentially means that all new transactions must be properly checked to ensure the block chain recognizes that none of the transaction inputs have not been spent (Oluwoye 9). Certain outlets known as "miners" (other peers on the network) exist on the Bitcoin network and they are the ones that offer the computational power to take on load that is required in validating

Bitcoin network ledgers/transactions. Once they have been validated, the transactions are then grouped up to be displayed as a hash tree or a Merkle tree, which will look something like this:

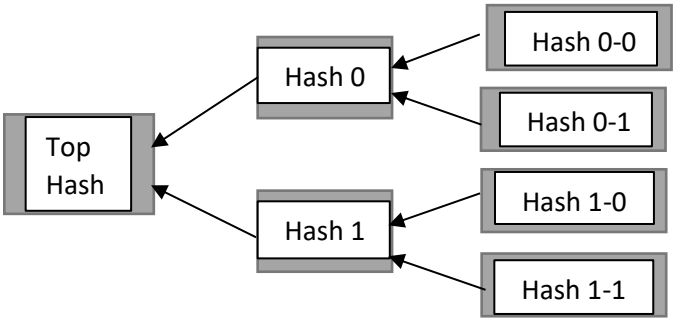

Figure 1. Merkle Tree.

These hash trees are included in their respective block headers along with other significant pieces of information. Inside the block header, information like time, version of software (for computers to read each block), the hash of the last block on the blockchain and a target value, will also be included. This target value is also called a nonce (random whole number), which is basically a dummy field or range, that miners use to help guide them adjust a block hash below the target

value. For the nonce to be successful, the block header must be continuously incremented and hashed (SHA-256 hash algorithm) until a block is returned that is less than the target value. Once the nonce is successful, the block is ready to be included onto the blockchain. The blockchain can be thought of as a list, the head block of this list is known as the "Genesis block." Each block points to the one behind it through the same hash code, starting from the genesis block. The network is constantly being updated every 10 minutes. Once a block has been added onto the network, it cannot be modified or altered in any way.

Proof of Work

As mentioned earlier, the process known as proof of work is needed in every block. This process is used to prevent certain vulnerabilities on the network, known as hash tampering. Before the verification process, the objective is to have a resulting hash under a 256-bit number. Therefore, the starting input for the hash functions is a random number that will hopefully result in a hash lower than the initial target 256-bit number. To begin this process on the network, a set of data is put through a hash function, in this case SHA-256 (Frankenfield 2021). It is important to note that one slight change in the data itself will result in a completely different and unrecognizable hash result. The size of the data will also be preserved upon conversion with the hash function (Frankenfield 2021). The significance of this concept lies in its process. Using SHA-256 to verify data only goes one way, therefore working backwards will not result in the original data (Frankenfield 2021). As miners undergo this process on their given block, it slows down the creation of new blocks. The duration of validating a block for any errors takes approximately 10 minutes (Frankenfield 2021). During this time, if a miner discovers that the block has been tampered with, it cannot be added to the Bitcoin network. If the miner discovers no issues with the block, it is then successfully added to the end of the blockchain as mentioned

previously. When this block has been added, the miner then receives their reward that was initially attached to the block that they have worked on.

Data Tampering

Once a block has been verified, the block chain ensures that the transaction record data cannot be altered. If the hash for a block were to change, every other block linked to the tampered one, must also be changed. This type of tampering can also be known as tampering backwards (Zhang, Xue and Liu, pg. 1:21, 2019). The most extreme case of this type of tampering involves what is known as a 51% attack. This attack refers to the hypothetical idea of owning majority control of a network, in order to interrupt the creation of new blocks by preventing other miners from completing/verifying blocks (Zhang, Xue and Liu, pg. 1:17, 2019). Since every block's hash must now be altered, it consumes a tremendous amount of computing power to do. As a reference, a Stack Exchange user in 2011 was curious as to how much computer consumption it would take to launch a 51% attack (Schwartz 2011). A user named David Schwartz replied with the following hypothetical:

a. A standard laptop CPU provides around 2-8 Mhash per second.
b. A traditional ATI 5870 graphics card which happened to have the best efficiency per cost ratio at the time, mines at about 400 Mhash per second.
c. The Bitcoin network at that time had a total hashing power of about 12,460, 000 Mhash per second.
d. One ATI 5870 graphics card had cost around $290 dollars.

Based on the specifications detailed above, one would need about $10 million dollars for video card purchases in order to acquire the hashing power of about 12, 460, 000 Mhash per second,

that is needed to pull of this hypothetical attack. Not to mention, the cost of assembly and other hardware that can arise from device issues, repairs, and replacements (Schwartz 2011). If one was able to cover these costs and obtain the materials to proceed with this experiment, altering each block is time consuming. To alter one block's hash, every block that is linked to that block must also be changed, as they become invalidated. As this requires a lot of computing power, this can put a strain on equipment and can be very time consuming and not very efficient. As David Schwartz also mentioned in his reply on StackExchange, if the price of Bitcoin continues to increase which has been happening, the consumption costs will also rise. Ultimately, data tampering within Bitcoin is feasible but it is very difficult and almost impossible to pull off.

4 ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM

As blocks complete proof of work, they must verify each transaction before adding them onto the network. An algorithm known as the Elliptic Curve Digital Signature Algorithm is used to verify where a transaction is coming from. This algorithm essentially builds up from the idea behind another algorithm known as the digital signature algorithm. A digital signature is basically a modified value that is a result of some mathematical equation, that is also used to validate the integrity and security of a message or piece of data. The main ideas behind the digital signature algorithm include, both a private and public key exist, “the signer” (original sender) would use the private key and their original message to generate a digital signature, and that the receiver would use the public key with the digital signature to verify and access the original message. Going back to ECDSA, this algorithm is bounded by how many bits are used in the actual equation (Alaoui 2012). Bits are a computer’s way of representing data and are often expressed with zeros and ones, which is known as binary notation. Eight bits are also referred to as a byte. Every time it is incremented, the range of available values double.

-	Number of bits:	1	2	3	4	5	6	…
-	Range value options:	2	4	8	16	32	64	2^n

Thirty-two bits has a range value of 4,294,967,296, which already is a pretty big number for the algorithm (Alaoui 2012). ECDSA typically uses 160 bits, therefore the range of values will also be a substantial and time-consuming number that can be represented by 2^{160} (Alaoui 2012). The algorithm also uses a SHA-1 cryptographic hash to sign (or verify) a message/data. Hashes are mathematical equations that will be applied to every byte of data and will return a unique number or character to your message (Alaoui 2012). SHA-1 will consist of 20 bytes since the ECDSA is composed of 160 bits (Alaoui 2012). One slight altercation in the original message or data file will cause the hash to change completely, resulting in the invalidation of the signature and the algorithm.

Elliptic Curve Cryptography

Before proceeding with understanding how the ECDSA is used, elliptic curve cryptography must first be examined. This type of public-key cryptography is based on the notion of a trapdoor function. This means that given 2 values for A and B, the "trapdoor" function can be used to go from A to B. Although the initial points are randomly selected from an elliptic curve similar to the one portrayed in figure 2. On the flipside, it is very difficult to go backwards from B to A (Wagnon 01:10-01:53). The curve that is used in this type of cryptography can be represented by the following graph:

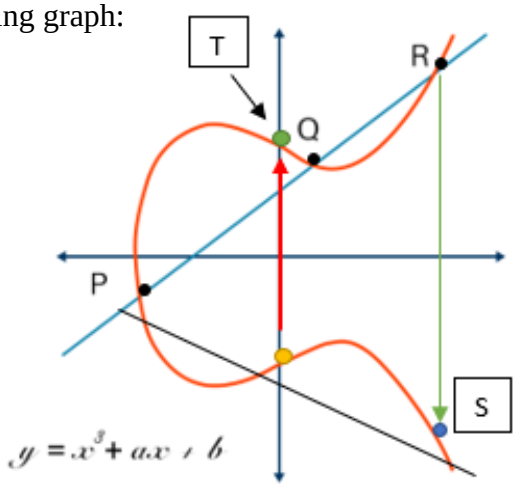

Figure 2. Elliptic Curve Sample

To begin, we select any point on the graph, we will use Point P as our starting point. Then, you must take the tangent line of Point P and that will result in two more intersections on the curve, as illustrated by Point Q and R. Next, we have a way to "add" these points together on the graph, which is called the "dot" function. If we take Point P and use the dot function with itself, it yields the straight line represented above. Furthermore, if we take Point P and apply the dot function with Point Q, it will yield Point R. If we continue using the dot function, the result can be found opposite of Point R, on the x-axis. We are then able to connect a line from Point P to Point S, which will result in a new intersection. If we continue the dot function once more, the result will be found opposite of the yellow intersection point, on the x-axis. This will again, result in being able to draw a line from the starting point to our new Point T. This cycle can be continued "n" number of times, which will also serve as our private key. It must also be noted that there exists a max-bound in the x direction, which is also equal to the public-key size (Wagnon 07:28-08:28). Therefore, the greater the key size, the more availability in points on the graph and a greater number of dot function cycles can occur (Wagnon 08:28-08:55). This results in an increase of values available to use as points on this curve. Ultimately, the greater the bounds, the more difficult it is to work backwards and reach the beginning value.

5 BITCOIN ADDRESS

After the verification and signature of messages on the Bitcoin network, the network must then be able identify the addresses or wallets involved in the transaction. Bitcoin wallets essentially enable the sending and receiving of cryptocurrency by adhering to the Bitcoin protocol. The idea behind owning cryptocurrency can be simplified down to the following: you do not actually own the specific Bitcoin that you hold. Instead, you own the combinations of the keys that allows you to access the Bitcoin and move it around. Bitcoin addresses for the most

part, either begin with a 1 or a 3. All Bitcoin addresses that start with a 1 are known as a SegWit wallet. These types of wallets follow what is known as the Segregated Witness Protocol. This protocol updates the Bitcoin by making transaction sizes smaller, which allows the network to handle more transactions at once. This is achieved by separating Bitcoin signature data from transaction data. SegWit wallets will primarily follow one of 3 address formats (Sedgwick 2019). The first address format is known as P2PKH (Pay-to-Pubkey Hash) or legacy addresses and they all typically begin with the number 1 (Sedgwick 2019). This type of address is Bitcoin's original address format which is still reliable and fully operational to this day. The second type of address format is known as P2SH (Pay-to-Script Hash) and it begins with the number 3. Its format is also structured similarly to P2PKH's format. Not to mention, this also happens to enable more functionality than legacy addresses (Sedgwick 2019). This type of address is widely supported and can be used to send funds to both P2PKH and bech32 addresses. If your Bitcoin wallet does not begin with a 1 or a 3, then it will most likely begin with the prefix "bc1", which belongs to the last address format. All addresses that begin with this prefix, follow the bech32 format and are longer than legacy addresses. Bech32 addresses are also native SegWit format and are supported by majority of software and hardware wallets, but a minority of exchanges (Sedgwick 2019). For example, Ledger and Keepkey wallets allow for the sending of funds to Bech32 addresses but do not allow users to receive them with this format (Sedgwick 2019). As of recent, less than 1% of Bitcoin is stored in Bech32 addresses, although this number appears to be slowly increasing.

Generating Bitcoin Wallets

The generation of Bitcoin wallets relies on the idea of public key cryptography. This includes private keys and public keys that are only known to the owner/users (Tore 2020). To

generate a traditional Bitcoin address, the public key will undergo several hashes to reach a combination of letters and numbers that is safe and secure enough to serve as a digital wallet. After obtaining a public key from the previous implementations, the key is then hashed with the SHA-256 algorithm, resulting in a slightly shorter combination of letters and numbers (Tore 2020). The next step involves hashing the resulting combination with the algorithm RIPEMD-160, which will result in an even shorter combination of letters and numbers (Tore 2020). This result is then hashed with SHA-256 two more times. With this output, it is crucial to remember the first 4 bytes or 8 characters/digits from this combination. These 4 bytes must then be added to the end of the output that was given from hashing with the RIPEMD algorithm (Tore 2020). With this new combination of letters and numbers, it is then finally converted into a Bitcoin address format using a Base58 function (Tore 2020). This new series of letters and numbers is secure enough to use. With this wallet, a user can be fully anonymous and ready to handle any sort of transaction.

6 VULNERABILITIES

By storing funds in these digital wallets, users must accept the risks that are involved with this system. Because transactions include a direct path from consumer to consumer and skips out on involving intermediaries, failed or incorrect exchanges cannot be resolved with third parties such as banks or credit card companies. Users are fully anonymous therefore, the person exchanging their currency must be certain on where (Bitcoin address) they are directing their funds. It is important to note that by even accessing an internet connection, a user is at risk. Users must be attentive to internet security since hackers can essentially get access to a user's Bitcoin wallet simply by breaching their computer's internet connection (Latifa 15). The lack of third-party regulation over this system certainly makes users more accountable for their actions

and assets (Latifa 15). While the technology behind cryptocurrency continues to evolve each day, it is crucial to note that attacks on Bitcoin are not impossible; however, they can be very expensive.

Double Spending

One type of attack that is uncommon yet still occurs is known as double spending. It ultimately persists of the Bitcoin network being interrupted by disguising real and fake transactions together. The issue arises when the network needs to check and confirm the transaction. By the time the network needs to perform this check, another user or the same user can broadcast the same transaction onto the network (InnerQuest 2018). The problem results from the network being given the same transaction, therefore implying that the system recognizes two different transactions occurring and performing an exchange twice. This leaves a user with an unwanted transaction and most likely an additional and unwanted charge of another real transaction that occurred not so long ago. This then leads to the question; how can the network distinguish between real and fake transactions? This ultimately goes back to the system upon which Bitcoin was created. With Bitcoin, their solution to this issue relies on the majority of nodes (blocks) on the network. If they agree on which transaction was first to be received, later attempts to double-spend are irrelevant (InnerQuest 2018). It is because of Bitcoin's system where they timestamp transactions, that once transactions are verified, altering or double spending is almost impossible.

51% Attack

Another rare exploitation for Bitcoins includes the 51% attack. This attack refers to the hypothetical idea of owning or controlling a majority of a network's computing power. By

owning 51% of the network, the attackers would be able to prevent new transactions from getting confirmed, which then leads to a halt of payments between some or all users (Frankenfield 2019). This would also allow for the possibility of reversing transactions that were already completed, as long as the attackers were still in control of the network at the time. In other words, this type of exploitation can also be described as double spending. By preventing other miners from creating blocks, attackers can theoretically monopolize the mining of new blocks and earn all the rewards (Frankenfield 2019). The most recent recorded example of this attack was back in May of 2018. At the time, a majority of Bitcoin Gold's (another cryptocurrency) network had been hijacked which then led to the raising the exchange thresholds of that coin. The attackers were able to double-spend for several days, eventually stealing more than $18 million worth of Bitcoin Gold (Frankenfield 2019).

Timejacking Attack

The Timejacking attack focuses primarily on creating double-spending transactions, leading to a delay of computational resources by altering the network's time counter. Essentially, every node or block on the network has a timestamp derived from the time upon which its peers connect to the network. When a node connects to the network for the first time, the median time of its peers is sent in the version message (FSA, pg. 62 2018) The version message simply acts as a carrier of information in which a timestamp is included of the initial connection to the network. If the median time deviates from the system time by more than 70 minutes, however, the network time counter reverts to the system time (FSA, pg. 62 2018). This network time is also required when verifying new blocks. Upon creation, it is crucial that the network time is behind the timestamp located inside the version message. If any new block's timestamp is ahead of the current network time by more than 120 minutes, it is rejected by nodes (FSA, pg. 62 2018). A

block whose timestamp is earlier than the median time of the past 11 blocks is also rejected (FSA, pg. 62 2018). This verification puts upper and lower bounds on the acceptable range of block timestamps (FSA, pg. 62 2018). Ultimately, this can lead to the possibility of speeding up or slowing down a certain node's time counter by transmitting a forged version message. This vulnerability is precisely what this attack focuses on and manipulates.

Spam Transactions

While the previous attacks were strictly inside the Bitcoin network, there also exists explorations that exist surrounding human error/misunderstanding. This would include a simple scheme involving spam messaging or sending large amounts of unwanted affiliate advertising/marketing. Typically, these large amounts of transactions are small but when grouped together in significant numbers, could cause a backlog within the network. Due to the large traffic the network receives all at once by this spam, it creates the possibility of congestion for the miners, leading to the backlog of transactions mentioned previously.

7 DISCUSSION

With all the algorithms that are involved during the blockchain process, it is clear that no piece of data is easily accessible. With how unrecognizable the results are from hash functions to the originals data, the several mutations that occur in this process surely do provide a decent amount of security. Another significant security point is included inside the ECDSA process. As mentioned previously in that section, the algorithm is bounded by how many bits are used. Since ECDSA is typically 160 bits, that means there exists $1.461501637 \times 10^{48}$ (2^{160}) possibilities to chose from as starting points. Therefore, the higher the bounds used; the more computational power required crack the algorithm. As previously referenced in the data tampering section, the

hypothetical computer specifications proved to be very costly in administering an attack on the Bitcoin network. While Bitcoin is not completely flawless, it is entry points to attacks seem very expensive. The requirement for computational power can prove to be very overwhelming for the typical person, moneywise, resource wise and timewise. The continuous increase in Bitcoin's value and transactions also play a significant role in creating a bigger network and an even bigger problem for attackers to deal with.

8 CONCLUSION

New cryptocurrencies continue to flood the market in hopes of being the next Bitcoin. While Bitcoin's value fluctuates daily, its popularity is continuously increasing. This has also led to an increase in demand for Bitcoin to be recognized as a form of payment in our economy. Questions begin to arise around the reliability of Bitcoin's security. While Bitcoin is not completely faultless, its structure provides several points of good enough security for its users. As indicated throughout the paper, every aspect within Bitcoin can be targeted but it comes with a hefty price. The material and computational cost continues to increase significantly. Therefore, at what point is it still even worth it to attack the Bitcoin network? It's safe to say that while attacks on the Bitcoin network are becoming less frequent over the years, its security is stable enough to match its increase of value, users, and demand.

9 REFERENCES

Latifa, Er-Rajy, et al. "BLOCKCHAIN: BITCOIN WALLET CRYPTOGRAPHY SECURITY, CHALLENGES AND COUNTERMEASURES." *Journal of Internet Banking and Commerce* 22.3 (2017): 1-29. *ProQuest.* Web. 23 Oct. 2020.

Oluwoye, Oyedeji A. "Digital Cryptocurrencies: The Design and Network Analysis of the Bitcoin Infrastructure." Order No. 10189085 Alabama Agricultural and Mechanical University, 2016. Ann Arbor: *ProQuest.* Web. 23 Oct. 2020.

Iansiti and Karim R. Lakhani, Marco, and Karim R Lakhani. "The Truth About Blockchain." *Harvard Business Review*, 21 Aug. 2019, hbr.org/2017/01/the-truth-about-blockchain.

Zhang, Rui and Xue, Rui and Liu, Ling. "Security and Privacy on Blockchain." *Association for Computing Machinery*, Surv.1, 1, Article 1 (2019), https://arxiv.org/pdf/1903.07602.pdf

Sedgwick, Kai. "Everything You should Know About Bitcoin Address Formats." *Bitcoin.com*, 18 Feb. 2019, https://news.bitcoin.com/everything-you-should-know-about-bitcoin-address-formats/

David Schwartz (David-Schwartz), "How many consumer computers would it take to launch a 51% attack?" #1056. 14 Sept. 2011. Forum post. https://bitcoin.stackexchange.com/questions/1056/how-many-consumer-computers-would-it-take-to-launch-a-51-attack/

Alaoui, Youness. "How the ECDSA Algorithm Works." *KaKaRoTo's Blog*, 31 Jan. 2012, kakaroto.ca/2012/01/how-the-ecdsa-algorithm-works/.

Wagnon, John. "Elliptic Curve Cryptography Overview." *YouTube*, uploaded by F5 DevCentral, 14 Oct. 2015, https://www.youtube.com/watch?v=dCvB-mhkT0w

Seguias, Bassam El Khoury. “Bitcoin Elliptic Curve Digital Signature Algorithm (ECDSA).” (2018). https://delfr.com/wp-content/uploads/2018/12/Bitcoin_ECDSA.pdf

Tore, Tuna. “How to generate a Bitcoin address – Technical address generation explanation and online course.” *Medium*, 2 Jan. 2020. https://medium.com/@tunatore/how-to-generate-bitcoin-addresses-technical-address-generation-explanation-and-online-course-a6b46a2fe866

Team InnerQuest Online. “How Does a Blockchain Prevent Double-Spending of Bitcoins?” *Medium*, 25 Aug. 2018. https://medium.com/innerquest-online/how-does-a-blockchain-prevent-double-spending-of-bitcoins-fa0ecf9849f7

Frankenfield, Jake. “51% Attack” *Investopedia*, 6 May. 2019. https://www.investopedia.com/terms/1/51-attack.asp

Financial Services Agency, Japan. “Research on Financial Transactions Using Distributed Ledger Technology, Research on Technological Risks in Financial Transactions Using Blockchain.” March 2018. https://www.fsa.go.jp/en/policy/bgin/ResearchPaper_ISID.pdf