The Blockchain and How it Works

by

Sulaman A. Yaqoob

Submitted to the Department of Mathematics/Computer Science

School of SUNY Purchase

in partial fulfillment of the requirements

for the degree of Bachelor of Arts

Purchase College

State University of New York

December 2021

Sponsor: Professor Knarik Tunyan

Second Reader: Professor Alan Anderson

Table of Contents

Abstract

The purpose of this research was to delve into how a blockchain algorithm is formed and operates as it does within the world of cryptocurrency. This new and rising technology has revolutionized the world of currency and commerce and is finding its way into all kinds of industries which affect us all. Understanding its key components and history was essential and the information was used to produce a cryptocurrency coin in the Python programming language myself and simulate real life transactions. From executing the simulator program myself I came to understand how and why an algorithm like blockchain is so effective and secure. Its hash values, the basic unit of each block in the chain is what makes it nigh impossible to reverse engineer, and if you manage to somehow do it and change the hash, the true current hash would be so radically different that the system or even the human eye can notice that the chain has been tampered with and any transaction would not go through so long as the chain has a most current hash value (which it always does). A broken chain would just not be feasible nor work if you do not already have the proper sequence of hash values. An in-depth understanding of the blockchain is necessary to build a secure network for any digital currency and beyond.

1. Introduction

Within the field of cryptography, the algorithms which run cryptocurrencies utilizes blockchains. Within blockchains there are certain components which make it up, the most basic unit being the hash value. Being able to convert to hash values can eventually lead up to making your own blockchain and potentially something like your own cryptocurrency if you want. The genesis and origins of the idea of blockchain and how it developed till the modern implementation is of key importance to understanding why it was developed and for what it can be utilized for. The question for my senior thesis is, what is the blockchain's history and uses, and how do you simulate a blockchain yourself?

1.1. History of Blockchain

Before delving into a topic right away, it is good to learn about its history and how it developed through time till today to get a better grasp of the roots of the topic. When we hear cryptocurrency or blockchain we usually think of Satoshi Nakamoto (which is a pen name of the person or group) as the founder of the crypto-blockchain field. Whilst not entirely wrong (as Nakamoto was the first to practically implement blockchain), the concept of a blockchain and a digital currency actually goes back to the 1990's when Stuart Haber and W Scott Stornetta for the first time described a cryptographically secure chain of blocks (History of blockchain, 2019). Whilst working at Bellcore labs they were able to pick their own research project and the

problem of authenticating digital documents intrigued them. In other words, how could you be sure that you were looking at the original version or not? The main problem they ran into was that traditionally you would need to have trust, but trust made the problem insoluble. Even if there was a third party to authenticate it, there is a possibility of collusion there as well. So, the dilemma was such that if you kept needing outside parties to verify the authenticity, it would go on for Infinity. In an interview with both, Scott said, "I realised that if you turn that upside down and created a system of interlinked documents with essentially everyone as a witness, then you had in fact solved the problem" (Miller, 2021).

Stuart and Haber published their paper "How to Timestamp a Digital Document" published in 1991 made the argument that to properly verify digital documents was to timestamp them. They argued that because digital documents are easily tampered with, the method of timestamping was needed with two properties:

1. Timestamp the data itself regardless of the medium it appears on, so even if one bit changes it will be apparent.
2. Impossible to stamp data with a different time and date than the original one (Haber, 439).

After having come up with the idea of timestamping digital documents, they described the all-important hash function algorithm to be used to properly secure the document. If you are not sure what a hash value is, it will be talked about in detail later. But from Stuart and Haber's

paper, one part of their hash section stands out which will prove beneficial for understanding how their findings relate to the modern blockchain. Here is a snippet from their work:

1. The TSS sends our client the signed certificate $s = \sigma(C_n)$, where the certificate

$$C_n = (n, t_n, \mathrm{ID}_n, y_n; L_n)$$

consists of the sequence number n, the time t_n, the client number ID_n and the hash value y_n from the request, and certain *linking information*, which comes from the previously issued certificate: $L_n = (t_{n-1}, \mathrm{ID}_{n-1}, y_{n-1}, H(L_{n-1}))$.

2. When the next request has been processed, the TSS sends our client the identification number ID_{n+1} for that next request.

Having received s and ID_{n+1} from the TSS, she checks that s is a valid signature of a good certificate, i.e. one that is of the correct form $(n, t, \mathrm{ID}_n, y_n; L_n)$, containing the correct time t.

(Haber, 444)

After having shown how they derived the algorithm, this snippet is an explanation of how it is implemented in theory. This format of variables representing the sequence, time, ID, hash value, etc. is more or less similar to what bitcoin and other cryptocurrencies use when programming their currencies. It is the format that my own simulation will use later in this paper. Keep in mind that Stuart and Haber had a broader vision of the timestamp than just currency and hence have variables that may not be needed for currency strictly, but their method and base is what is applied today (Miller, 2021).

1.2. Szabo's Bit Gold

In 1998 the computer scientist Nick Szabo made one of the earliest attempts at making a decentralized digital/virtual currency called "Bit Gold" (History of blockchain, 2019). Bit Gold was never implemented, but besides similarity in name it is regarded to be the precursor to Nakamoto's Bitcoin protocol. Both their protocols have such similarity that there is speculation that Szabo is the one using Nakamoto as a pseudonym, though Szabo has denied this theory himself (Bit Gold, 2021). However, in 2008 before the bitcoin whitepaper was published, Szabo did comment on his blog that he was making a practical version of his hypothetical currency (Bit Gold, 2021), which still leaves a big question mark on Nakamoto's identity. Bit Gold uses cryptography and mining to establish decentralization. Elements such as Stuart and Haber's timestamped blocks that are stored in a title registry then generated using proof-of-work (PoW) strings (Bit Gold, 2021). PoW is a system the requires feasible amounts of effort to deter ill intended uses of computing power like spam emails or denial of service attacks (Proof of Work, 2021). Though many similarities between Bit Gold and Bitcoin, there are also clear differences than what Szabo intended with Bit Gold. The goal of Bit Gold was to address a similar issue that Stuart and Haber addressed as well, trust. Bit Gold (2021) states:

> Szabo said he created bit gold to address some of the inefficiencies in the traditional financial system. According to Szabo, parties must invest a great deal of trust in order for transactions to take place in the traditional financial system. For example, when a consumer wants to take out a loan, they first must locate a broker. Then, once they've

> accepted the loan from a financial institution, the institution must trust that that individual will repay the loan as agreed. By the same token, customers of a bank must trust that their money is well-secured and not being embezzled by the bank.

Szabo said himself at the 2015 Bitcoin Investor Conference the purpose of Bit Gold was to, "minimize vulnerabilities of all parties to each other" (Bit Gold, 2021). The blockchain as we can see, has from its origin been about security and trust.

1.3. Nakamoto's Bitcoin

Now that some of the fundamental history and due credit to the development of blockchain has been established, the one who practically implemented the blockchain and revolutionized the world of not just digital currency but of currency in general, was none other than Satoshi Nakamoto and his Bitcoin. Bitcoin was released in 2009 officially but its real take off and claim to fame was in 2014 when its value shot up and everyone started taking it seriously. In the early days, everyone was conversing about its legitimacy or how it went up so much, the key innovation it brought with it, the "blockchain" was overlooked (Phillips, 2014). The blockchain went live with Bitcoin and the blockchain is a public ledger of all Bitcoin transactions (or any cryptocurrency for that matter). When transactions occur, they are verified on a peer-to-peer network in which computers running bitcoin software (also known as nodes) reach a consensus, prevents fraud, double spending, or forgery. The key component as we know is the blockchain which keeps Bitcoin and any cryptocurrency transparent, secure, and

decentralized; ensuring all those who own a Bitcoin have a copy of the blockchain (public ledger) and hence all transactions are recognized and verifiable (Phillips, 2014) because of the flow of the blockchains which is undisrupted holding the hash values of the previous transaction, creating a unique and irreversible hash value for your transaction.

One of the main problems that Stuart, Haber, and Szabo sought to fix was the issue of trust. Bitcoin did just that because it removed the need for a third-party. You can transfer money using cryptocurrency without any intermediary using the blockchain. According to Alex Mizrahi the founder and lead developer of ChromaWallet, “As far as I know, the block chain is the only method to achieve global consensus which isn’t based on trust [of an external organization]” (Phillips, 2014).

Despite being the gold standard of the crypto world and hotshot it is today, the very first real-world transaction of Bitcoin (or any cryptocurrency) took place on 22 May 2010, when Laszlo Hanyecz bought two Papa John pizzas in Jacksonville, Florida for 10,000 Bitcoin. An amount which would be more than $600 million if held in last year.

2. Applications of the Blockchain

The application for blockchain thus far has been discussed in the context of currency, but as mentioned in section 1.1, the vision for blockchain was much broader than just for currency transaction. As time goes on from the release of Bitcoin, the blockchain is starting to branch out and the technology's potential is beginning to be used in other ways.

Ethereum, which is another cryptographic ledger, goes beyond even the concept of the blockchain. It offers a platform and programming language which can decentralize and secure as well as trade: domain names, financial exchanges, voting, crowd funding, company governance, contracts, intellectual property, and smart property (Phillips, 2014). The applications of the blockchain are numerous and counting, its potential has just begun to be realized.

2.1. Food Safety

With growing concern over food and drink safety and authenticity, sellers and manufacturers of food were looking for ways to be more transparent and trusted by having evidence of the life cycles of the food people eat. The Food Standards Agency (FSA) employed blockchain in 2018 at a cattle slaughterhouse. It was the first time blockchain was used in the food sector as a regulatory tool (Paul, 2020). Since its success, it is now applied in other areas of the UK's food industry. Farmers often record the many processes they perform like shed cleaning to crop spraying and harvesting, but on paper which is time consuming and outdated.

Such is the case with the oat farmers of Britain. But with blockchain, it enables all those growing, verifying, and assuring the growth of the oats to add any digital data throughout the oat's lifecycle. How that would work is any farmer working on the oats for instance could add digital data to the shared but encrypted register in a controlled and secure process, based on the permissions the individual has been granted to access (Paul, 2020).

2.2. Health Records

As we know, blockchain transfers data in a very secure and third-party free manner. This has huge implications for the health industry where large amounts of medical data must be stored and shared, many times across the globe. On top of that, privacy and security is a priority of this information, the "trust" which is put in the hands of the medical industry is arguably the most sensitive as it has all of our private medical data. The use of blockchain has many potential benefits for health records but also pitfalls. A potential way it can be implemented is that patients will basically own their own health records and then allow or deny access to doctors or anyone which would prevent any unauthorized access, tampering, or sale of information to a third party (Paul, 2020). There is a problem to consider however, and that is due to the scalability of blockchain into an already complicated and messy system, and how private the stored data is. According to Cosima Gretton, clinical product manager at Mindstrong Health teaching at University College London, "Data on the blockchain is accessible by anyone, and storing and computing data on the blockchain is slow and expensive" (Paul, 2020). The Enigma Project may

be a solution to this problem, however. It is an off-chain network which acts as an extension to blockchain platforms. Gretton then added in the same article:

> It allows code to be processed both publicly on the blockchain, maintaining a public ledger of a transaction, and on Enigma's off-chain network where the data is encrypted. By processing data off-network, the Enigma network can process-intensive computations that remain publicly verifiable on the blockchain.

Further supporting blockchain in the health industry is Imogen Farham, a researcher Reform, explaining that "What blockchain does well is present a way forward to transform the relationship patients have with their healthcare data. To achieve this, standardising how healthcare data is formatted to facilitate meaningful interoperability between systems would be a good place to start." (Paul, 2020)

2.3. Gaming

The blockchain being digital in nature itself, one would think it would be a perfect addition to the virtual world of games. And according to Forte, an economic technology building company for games, "The technology's unique properties map perfectly to the digital nature of games and enable the secure ownership of in-game digital assets, thereby laying the foundation for ground-breaking blockchain- and crypto-native marketplaces and services for the games industry" (Paul, 2020). With video games advancing and concepts like in game currencies and

purchases already popular and gaining traction, blockchain could be applied in a multitude of ways for buying, selling, and managing in game currency accounts. Companies like Ubisoft and Azarus have already been using blockchain for users to complete challenges using the EOSIO blockchain protocol. They watch streams and get credits for watching whilst the credits are redeemable for in game items. A user can make their own rules and challenges using the platform, promoting transparency and fairness.

Gambling games is a great avenue for blockchain to enter because virtual gambling at the moment lacks transparency, regulation control, and players can be easily taken advantage of by the companies. Blockchain can add that security and safety making sure that results and outcomes are as intended without any interference or trust issues.

EOSIO is a cryptocurrency platform whose team is building a blockchain network for the video game industry. Their project is called Ultra, which is a blockchain-powered video game ecosystem which allows players and developers with new and unique tools and functionalities (The Ultra Blockchain, 2021). Ultra Games is the application they are working on creating, which is a distribution platform like Steam, but will be able to provide features like game reselling, virtual item creations, trading, referral systems, and more, all powered by the EOSIO blockchain (The Ultra Blockchain, 2021). This is a very interesting and innovative undertaking, a Steam like platform but all based on its own currency and blockchain which allows it to do things which Steam cannot, and some of the features Ultra can provide are desired by Steam and other game platform users as well. This platform will not only have an impact on new features available to users, but the app developers have said that compared to the 30% Steam charges

developers on their game selling, Ultra can charge at 15% (The Ultra Blockchain, 2021). Which is a significant competitive advantage off the bat. Gaming is and will be greatly impacted by the introduction and fine tuning of the blockchain in the coming future, it is only a matter of time it seems before it becomes the norm.

2.4 Smart Contracts

Smart contracts simply put are programs that run on a blockchain when certain conditions are met. It essentially executes agreements of all the parties involved without a third party needed, hence it is much faster, efficient, and accurate as there is no loss in time or misunderstanding between intermediaries. These contracts follow “if/when…then…” statements written into code on a blockchain and then the network executes the actions when the conditions have been met (What are Smart Contracts, 2022). Possible commands and actions to be executed include, releasing funds, registering vehicles, sending notifications, issuing tickets, etc. and afterwards the data is uploaded to the blockchain when the transaction is complete. Thus, the transaction can not be tampered with and only the individuals or parties involved who have the granted permission can see the contract and results. The smart contract can be modified by a developer depending on the needs or stipulations to be satisfactory, but eventually templates and interfaces amongst other tools will simplify in structuring these contracts (What are Smart Contracts, 2022).

Other than the efficiency and speed of smart contracts, the trust and transparency of it is very alluring. Because blockchain records are not only encrypted but also connected to the subsequent records on a distributed ledger, anyone or any hacker would have to alter the entire chain to change a single record; which is why the blockchain is infeasible to reverse, meaning almost impossible to alter. This topic will be dealt with much more detail later.

3. Pros and Cons of Blockchain

Thus far we have talked about what the blockchain is and how its use could be beneficial for us, but there are legitimate arguments against the use and/or effectiveness of blockchain. Any new technology which is released has a stigma around it whether it be on its general use or in specific circumstances, which is a natural response to anything new and untested. But keep in mind, everything is bound to have pros and cons even mainstream technology. With that, lets go over the advantages and disadvantages of blockchain. As the pros have been mentioned quite frequently earlier, they will be mentioned briefly as a reminder. The cons will be the focus of this section.

3.1. Pros of Blockchain

- Disintermediation: No third party will be needed in verifying or authenticating hence delay times and possible tampering are avoided. Also, any fees one incurs using a middleman will cease, saving money.

- High Quality Data: Blockchain ledgers are a consensus process which filters out and bad data with useful data automatically. No one can add just any data or manipulate it, it will even fix and eliminate any false data. Everything entered will be verified before added to the ledger (Iredale, 2021).

- Integrity/Security: The level of accuracy blockchain offers is arguably the best so far. The data you see will always be the right one because no one can alter them. As mentioned earlier, it is near impossible to reverse engineer and change a block since the entire chain would need to be altered, and the required computing power quite frankly is not practical

- Faster Transactions: The centralized traditional system usually through banks can take time to process a transaction, and even more for wire transfers or sending overseas. With blockchain, transactions anywhere on the globe can be done within seconds.

3.2. Cons of Blockchain

The cons will be dealt with in a bit more detail as they have not been mentioned yet. It is good to keep in mind that blockchain and its branches is still new and developing, so any cons or bugs can always be tweaked and polished. As the technology moves forward in implementation, the fields or areas in which it may lack will become more apparent and hence their solutions can also be found, as it is with any technology.

3.2.1. Redundant Energy Consumer

Redundant Performance is one of the cons, because when blockchain is updated, all of its nodes and servers update as well to have the ledger as the current version. Though this is a good thing, it is also computationally repetitive, thus undergoes the same process repeatedly (Iredale, 2021). Thus because of this consensus required by all the nodes needing to communicate back and forth to validate a transaction, a huge amount of effort and energy is needed.

3.2.2. Data Modification

This potential downside is quite a serious one. With the security and impregnability of the blockchain being great and all, it is not always optimal. If ever change is needed in the blockchain for whatever reason, it is very difficult and demanding requiring a hard fork method, where one chain is abandoned and a new one is taken up (Academy, 2021). The irreversibility

goes both ways, the hackers, and the legitimate users, so this extra security that blockchain provides could potentially prove a double-edged sword.

3.2.3. Inefficient

As efficient as it is, there is also an argument against the efficiency of blockchain. According to Yano (p. 54), the trustless transactions come at the cost of efficiency. It further goes to say that because of the constant need of reaching consensus and replicating data across the entire network, which could possibly have faulty or even dishonest nodes makes blockchain networks fundamentally inefficient. Yano's *Blockchain and crypto currency building a high quality marketplace for crypto data* (2020) further argues that there is no reason to replace the centralized systems with only decentralized ones. Rather, blockchain should be utilized to keep centralized systems honest.

3.2.4. On-Chain Only

The trustless security blockchain provides is guaranteed on the chain, but it is important to note that such a guarantee is only on-chain. Any agreements made off-chain that go against what is on the blockchain, there is little the blockchain can do. Designing the network to minimize any sort of bribery or tampering can work but cannot eliminate the potential off-chain arrangements which in the human world occur all but too often. It is interesting to note, that even

on-chain the protocols are designed to tolerate up to 30% or even 50% of “dishonest” nodes without fundamental loss of network integrity (Yano, p. 54).

3.2.5. Storage

The ledgers blockchain utilizes can grow very large with time. Bitcoins blockchain currently requires around 200 GB of storage, hence the growth in the size is faster than the growth in hard drives; thus, the network risks losing nodes if ledgers become too large to download and store (Academy, 2021).

Just like all technologies, blockchain has its upsides and downsides but it provides unique advantages as well. The impact it has already made is reason enough to believe it is here to stay. The versatility of blockchain and its network which goes beyond currency still has a long way to go until mainstream adoption, but the integration has begun to penetrate multiple industries. It is expensive and tedious to do an overhaul outright for now, but as all technologies it will ease its way into daily uses.

4. Blockchain in Use

With blockchain entering so many industries, which companies actually use it? Many companies have started to incorporate blockchain protocol in their transactions and record keeping, many of these companies are some of the most well known and renowned companies in the world.

Walmart has been using blockchain for a long time actually. They use IBM's supply technology "Hyperledger Fabric" platform to back their supply chain processes. Walmart also plans to track their food stuff from the farmers to the customers and will allow customers to even check the origin before buying an item (Iredale, 2021a). It is a work in progress however. Other companies like Ford, DHL, Pfizer, Shell, multiple airways, and many more.

Other than the numerous financial and banking related companies, many retail/consumer companies allow customers to pay with cryptocurrency as well, which was the origin point of the blockchain itself. AT&T was the first mobile carrier to accept cryptocurrency payments in 2019. Starbucks is another one which allows transactions with crypto. Every month there is a tweet from a CEO of some major company who announces that they will be accepting certain cryptos which makes the crypto market shoot up. So there definitely is a gradual acceptance.

These transactions are done through various crypto wallets which are third party platforms which work with the company. After an individual downloads the wallet app they can transfer funds there from their crypto wallets and start using them like a card through your

phone. Another way to spend crypto is to connect a debit card to it, so it lets you spend it like cash. Different currencies have different cards available and are issued by the major credit card companies like Visa and MasterCard (Hayes, 2021). Compare it with paper money, you store it in a bank and spend it digitally through bank transfers, PayPal, online, etc. it is also the case with cryptocurrency. The “wallet” for crypto is software app based and all your currencies will be in it and can be accessed from anywhere.

5. Hash Value, the Building Block

Now that this paper has gone over some of the important history and applications of the blockchain, the crux of the matter is at hand, the building of a blockchain. All the principles and concepts mentioned in earlier sections will now be put to the test in practical view. The first step in building the blockchain is understanding all the parts of a blockchain sequence starting with the hash itself. The hash function or a cryptographic hash function (CHF) is a mathematical algorithm which Maps data of a size to a bit array of a fixed size which is the “hash value” (Hash Function, 2022). This function is nearly impossible to invert making it a one-way function. If made properly the only way to find a message which produces a given hash is to attempt a brute force search (trying to find all possible outputs until the solution is found) over the possible inputs and see if a match is produced, or you can use a rainbow table (which is a space time tradeoff meaning it uses less time but more storage) of matched hashes. The hash function is the basic unit tool of modern cryptography.

Commented [MOU1]: Introduce every acronym before using it in the text. The first time you use the term, put the acronym in parentheses after the full term. Thereafter, you can stick to using the acronym.

The ideal cryptographic hash function has the following main properties:

- it is deterministic, meaning that the same hash always results in the same message
- Computing the hash is quick for any message
- It is nearly impossible to invert a given hash value (to reverse the process that generated the given hash value)
- it is infeasible to find two different messages with the same hash value
- a small change to a message should change the hash value so extensively that a new hash value appears uncorrelated with the old hash value (avalanche effect)

(Mehta, 2020)

Example Input Texts	Hash Values Using SHA-1
Hello	D364965C90C53DBF14064B9AF4BAABCA72196E2E
Hello! You are reading an article about the cryptographic hash function!	B26BACAB73C46D844CABEC26CE32B030FED1164F

(Mehta, 2020)

Commented [MOU2]: If this is a screenshot, add the reference below the figure. Refer to the resources on Senior Seminar Moodle page on how to format it.

This is an example of the hashing function at work. As you can see, each message is encoded, and even the slightest of change can alter the code so drastically that it will seem like the two have no relevance to each other. One of the most fundamental properties of cryptographic algorithms is that they should be extremely difficult to reverse engineer in order to find the input but should be easy to verify the output. Using the SHA-256 hash above, it is trivial for someone to apply the SHA-256 algorithm to “Hello World” to realize that it outputs the same

hash, however at the same time it should be very hard to take the output hash and get the input "Hello World" from it. Hence this form of cryptography is called one way. Bitcoin uses Double SHA-256, which is simply applying SHA-256 again to the SHA-256 hash of "Hello world". How Bitcoin does this is use SHA-256(SHA-256(x)) proposed by Ferguson and Schneier in their book "Practical Cryptography" later renamed "Cryptography Engineering" (Hashing or encrypting, 2011). Why the creators of Bitcoin used double SHA-256 is still debated, but it is usually agreed that it was used to further enhance security for its network.

Commented [MOU3]: Elaborate on this statement.

6. Irreversibility

As mentioned in the previous section, the "avalanche effect" is what occurs when the input is changed slightly but the output hash value is changed drastically. This image shows the avalanche effect visualized:

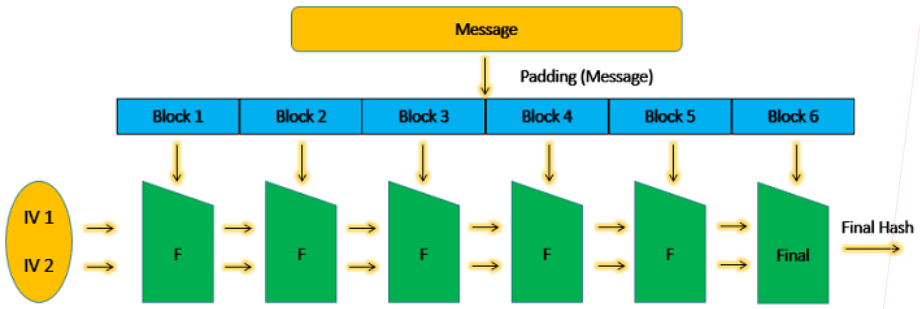

Image: The Avalanche Effect (the butterfly effect)

Commented [MOU4]: See above comment for formatting figures and screenshots

(Decoded: Examples of how hashing algorithms work, 2021)

According to the above picture, processing of the blocks occurs at the same time. The first blocks output is given as input along to the second block. Then the seconds data output to the third and so on. Hence the final output is the value of all the blocks combined. One change in the message at any point, the whole value changes and it will be clear that tampering has taken place somewhere along the chain.

In MD5 algorithm, here is a more apparent example of the avalanche effect at work:

```
MD5("The quick brown fox jumps over the lazy dog")
= 9e107d9d372bb6826bd81d3542a419d6
```

Even a small change in the message will (with overwhelming probability) result in a mostly different hash:

```
MD5("The quick brown fox jumps over the lazy dog.")
= e4d909c290d0fb1ca068ffaddf22cbd0
```

Here we can see that all it took was a “.” period to change the entire value that it seems like there is no correlation between the two messages.

7. How it works, Theoretical

Before we get into the code, I want to explain the theory in a more practical way than above. Let's say our coin is called the SeniorCoin (SC):

SeniorCoin (SC) has some transactions:

t1: Prof. Tunyan sends Sulaman 2 SC

t2: Bob sends Rebecca 4.3 SC

t3: David sends Tony 3.2 SC

These can be stored in a block "B1". This is the initial block (also called genesis block) which has info about the transactions. Doesn't have to have additional hash info or basis.
It can have any base as the block will be hashed with a hash value output. When coding, we will be using the SHA256 to hash the values. Here we will just make it up. Now this is how the block of B1 will look:

B1 ("AAA/START", t1, t2, t3) -> 76fd89 (hash result of first block)

B2 will have other transactions, and other than having just transactions it will also have the hash of the previous block as basis 76fd89, **B2 (76fd89, t4, t5, t6)**. If you remember, this format should look familiar because of Stuart and Habers' hash model from section 1.1.

Imagine a function SeniorHash (in reality we will use SHA256) SeniorHash() will take the input of the transaction string which is in this case "Prof. Tunyan sends Sulaman 2 SC" for example. In a real traded blockchain it will not say this but rather a number or hash. So we append or concatenate the strings of "AAA/START", t1, t2, t3 and we get the output hash of B1 using the SeniorHash() function.

We do the same thing for **B2 ("76fd89", t4, t5, t6)** by concatenating the transactions with the previous hash and get the new hash output of B2. Let's say the next hash is 8923ff, B3. We can do this with any number of blocks. It resembles a reversed linked list except you are not pointing to the block hence the hash of B2 cannot be used to navigate to B1. That is because it is very hard to find out where a hash is coming from. Outputting hash is easy, but reversing is near impossible or at the very least intensely difficult. You can't navigate to B1 like a linked list, rather a blockchain is called such because we have info of the previous block in the form of a hash in the next block.

If Prof. Tunyan sent Sulaman 2.1 SC instead of 2 like she originally did, that would change the original hash of 76fd89 to something completely different it would be unrecognizable. The resulting hash being so different would come out as let's say fa57bb. In turn all other blocks B2 B3 etc. will change as well. And because of that we can verify only needing the last hash value from the blockchain because if anyone tries to pretend like something else happened or if someone tries to tamper with it, the hash value will be completely different than the last hash you will have, and it won't work. changing anything in the first hash we will get radically different hash values. All values change and the chain is disrupted. So that can't be

done, every future transaction is based on the previous hash so if anything isn't based on it then we will see that it won't be the right hash which they have hence the transaction will not be accepted because of manipulation.

8. How it Works, Methodical

Now that there is a better understanding of the inner works of the methodology of the blockchain, here is the code used for this project to simulate a blockchain in Python:

Commented [MOU5]: Indicate what programming language you use.

```
import hashlib

class SeniorCoinBlock: #will be one block in the SC (senior coin)

    def __init__(self, previous_block_hash, transaction_list): #initializer

        # We need a list of all transactions and hash of previous block

        # and then calculate a hash value for the current block

        self.previous_block_hash = previous_block_hash

        self.transaction_list = transaction_list

        #now to construct a data string
```

```
        self.block_data = "-".join(transaction_list) + "-" +
previous_block_hash

# combined with a separator, join everything from Tlist and add previous blocks hash
        self.block_hash = hashlib.sha256(self.block_data.encode()).hexdigest()

# this is the resulting hash, passing hexadecimal string.
```

This is the making and defining of the class of the Senior Coin.

```
class SeniorCoinBlock: #will be one block in the SC (senior coin)

    def __init__(self, previous_block_hash, transaction_list): #initializer

        # We need a list of all transactions and hash of previous block

        # and then calculate a hash value for the current block
```

The "__init__" is a method which is called when an object from a class is created, and it allows to initialize the attributes of the class. In other words, define its characteristics. We are initializing the SeniorCoinBlock (SCB) class with the attributes of retaining the hash value from the previous nth block and then transact upon it to calculate the hash value for the current nth block.

```
self.previous_block_hash = previous_block_hash

        self.transaction_list = transaction_list
```

The “self.” Is the instance of the class, which by using it gives access to the attributes and methods of the class.

```
#now to construct a data string

        self.block_data = "-".join(transaction_list) + "-" +
previous_block_hash

# combined with a separator, join everything from Tlist and add previous blocks hash
```

Now we construct the actual data string that we will see in the execution of the code in the terminal. This is formatting the output the end user will see, and it will make sense at the end if it is difficult for now. We are telling the program that when it runs, print the current blocks base and concatenate (join) it with the transactions taking place, and join it with the previous hash. If you remember the previous examples, **B2 (76fd89, t4, t5, t6)** this is what we are telling it to print out essentially.

```
self.block_hash = hashlib.sha256(self.block_data.encode()).hexdigest()
# this is the resulting hash, passing hexadecimal string.
```

Here we are simply telling it to run the SHA256 algorithm to output the hash value for the previous block.

We will move on with the code now:

```
# Transactions:
t1 = "Prof. Tunyan sends 2 SC to Sulaman"
t2 = "Bob sends 4.1 SC to Sulaman"
t3 = "Sulaman sends 3.2 SC to Bob"
t4 = "Tony sends 0.3 SC to Prof. Tunyan"
t5 = "Sulaman sends 1 SC to Rebecca"
t6 = "Sulaman sends 5.4 SC to Tony"
```

Here the transactions to take place have been hard-coded and stored as variables, all denoted with “t” and the order of the transaction. The transactions are string variables which we will be able to see written in the terminal.

```
initial_block = SeniorCoinBlock("Initial String", {t1, t2})


print(initial_block.block_data)
print(initial_block.block_hash)
```

Now the fun begins. Keep in mind that format of the B1, B2, B3 examples from here on out, as that will be the format you will keep seeing. The initial or genesis block is being programmed to output in the terminal to show the words “Initial String”, t1, t2. Think of it as **B1(Initial String, t1, t2)** just like in the previous examples.

The resulting output is:

Bob sends 4.1 SC to Sulaman-Prof. Tunyan sends 2 SC to Sulaman-Initial String

ce313ebaad0076d807c76751d6eee752308aefb0487590043cb841be5dbc8742

As we can see, it printed out in the format we expected, and using T1 and T2 created the first blockchain. Now the next blockchains will use the previous hash values for the base rather than a placeholder like “initial string”.

```
second_block = SeniorCoinBlock(initial_block.block_hash, {t3, t4}) #using hash of previous as basis


print(second_block.block_data)
print(second_block.block_hash)
```

Now the second block is set equal to the SCB as well, but now the base value will contain the initial blocks hash, as denoted by "initial_block.block_hash". And it will compute it with t3 and t4 to make the new hash value. It will look like this:

Tony sends 0.3 SC to Prof. Tunyan-Sulaman sends 3.2 SC to Bob-
ce313ebaad0076d807c76751d6eee752308aefb0487590043cb841be5dbc8742
1105ffab8d0baacfea673433ce649b7271c3daeb560800bf9e88d77da61efc86

Similar explanation here but notice how long the hash value is. It's not because the new hash is longer, it's because the first hash line is the previous hash value from the initial block which is used as the base. The new hash value is the bottom one starting with 1105. So now we see how the previous hash value is utilized.

```
third_block = SeniorCoinBlock(second_block.block_hash, {t5, t6}) #using hash of previous as basis


print(third_block.block_data)
print(third_block.block_hash)
```

And now finally for the third block we store the second blocks hash as the base and compute it with t5 and t6. The resulting output is:

Sulaman sends 1 SC to Rebecca-Sulaman sends 5.4 SC to Tony-
1105ffab8d0baacfea673433ce649b7271c3daeb560800bf9e88d77da61efc86
34e83c92c53b960babb934fc54d14c05bdb0f98beca1c9379b4c63abe2682dc7

9. Results

Once again, we see how the transaction between persons and amounts is noted, as well as the importance of the previous hash used as a base for the next block. All together it looks like this:

Bob sends 4.1 SC to Sulaman-Prof. Tunyan sends 2 SC to Sulaman-Initial String
ce313ebaad0076d807c76751d6eee752308aefb0487590043cb841be5dbc8742
Tony sends 0.3 SC to Prof. Tunyan-Sulaman sends 3.2 SC to Bob-
ce313ebaad0076d807c76751d6eee752308aefb0487590043cb841be5dbc8742
1105ffab8d0baacfea673433ce649b7271c3daeb560800bf9e88d77da61efc86
Sulaman sends 1 SC to Rebecca-Sulaman sends 5.4 SC to Tony-
1105ffab8d0baacfea673433ce649b7271c3daeb560800bf9e88d77da61efc86
34e83c92c53b960babb934fc54d14c05bdb0f98beca1c9379b4c63abe2682dc7

Very blobby. But I have been talking about how reversing the process is near impossible for a while now, so now that we saw a simulation of a blockchain at work, let's see what and how severe changing something can be to the blockchain as I kept mentioning.

```
t1 = "Prof. Tunyan sends 2 SC to Sulaman"
```

Let's change this value to 2.1 SC and see what happens to the hash values. Here is the result of that one change, compare to the values above:

Bob sends 4.1 SC to Sulaman-Prof. Tunyan sends 2.1 SC to Sulaman-Initial String

ebdcfc971c1852e8f664b85c9d5e5bb70f23ac7ad6154d2834f14b3af6336302

Tony sends 0.3 SC to Prof. Tunyan-Sulaman sends 3.2 SC to Bob-

ebdcfc971c1852e8f664b85c9d5e5bb70f23ac7ad6154d2834f14b3af6336302

225d2d373b0cf0d801b6d215128505593bc40d7213a14dc9c0b9b62fa7c08e16

Sulaman sends 1 SC to Rebecca-Sulaman sends 5.4 SC to Tony-

225d2d373b0cf0d801b6d215128505593bc40d7213a14dc9c0b9b62fa7c08e16

95ba2a0ee1df91c63b1d42c7692039040bf59a3b724e197bd3ede2af7f061036

The hash values in both are in bold to make it easy to compare, and as you can see that minute change truly distorted everything. What are the implications of this in real world use though? I understand it as such, crypto currency or anything that uses blockchain for transactions or codes is very secure because if anyone tries to hack it or change something in the blockchains path, it will be very visible because the flow and values of blockchains would be distorted and tampered with clearly. Hence the transaction or code will not work/go through because the system will detect something is wrong, as it will have the true current hash value.

10. Conclusion

Though this is the completion of my Senior Project, I do not see it as the completion of the Senior Coin. I plan to keep developing the project and my skills adding more complexity and variables to make the coin better. At least two ways I can do that is adding a timestamp noting what time the transaction took place, and the other how to put it to practical use in everyday living like mainstream cryptocurrency. The purpose of me doing this which was understanding how a blockchain works and is made has been sufficiently met, and I can only build off from this amazing project from here on out. From the basic hash value to putting everything together in a functional blockchain simulator, I have thoroughly understood the basis of how cryptocurrency operates in the real world. A very crucial thing the research and simulation did for me as I hope it did for the reader, is demystify the blockchain. The broader implications from my findings reinforce and support the claims and statements made about blockchain by the numerous sources cited in this paper, that blockchain technology is as secure as you can get for now and its here to stay. Seeing its secure and well thought out nature, I believe it will not only stay, but it will thrive and revolutionize the way the world interacts with each other. Maybe one day I can release a coin of my own based on the findings and experience I attained from this project.

Bibliography

Academy, B. (2021, August 25). *Blockchain Advantages and Disadvantages*. Binance Academy. https://academy.binance.com/en/articles/positives-and-negatives-of-blockchain

Bit Gold. (2021, October 24). Investopedia. https://www.investopedia.com/terms/b/bit-gold.asp

Fundamentals of data structures: Hashing - Wikibooks, open books for an open world. (2021). Fundamentals of Data Structures: Hashing. https://en.wikibooks.org/wiki/A-level_Computing/AQA/Paper_1/Fundamentals_of_data_structures/Hash_tables_and_hashing

Garbade, M. (2021, October 24). *How to Create Your Own Cryptocurrency Blockchain in Python*. Dzone.Com. https://dzone.com/articles/how-to-create-your-own-cryptocurrency-blockchain-i

Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology*. https://link.springer.com/content/pdf/10.1007%2F3-540-38424-3_32.pdf

Hashing or encrypting twice to increase security? (2011, September 25). Cryptography Stack Exchange. https://crypto.stackexchange.com/questions/779/hashing-or-encrypting-twice-to-increase-security/884#884

Hayes, A. (2021, December 8). *How to Spend Bitcoin and Other Cryptocurrencies*. Dough Roller. https://www.doughroller.net/investing/how-to-spend-bitcoin-and-other-cryptocurrencies/

History of blockchain. (2019). Technology | ICAEW. https://www.icaew.com/technical/technology/blockchain-and-cryptocurrency/blockchain-articles/what-is-blockchain/history

How to build your own blockchain for a financial product. (2021, June 14). Django Stars Blog. https://djangostars.com/blog/how-to-build-your-own-blockchain-for-a-financial-product/#header4

Iredale, G. (2021a, May 31). *List of Top 50 Companies Using Blockchain Technology*. 101 Blockchains. https://101blockchains.com/companies-using-blockchain-technology/

Iredale, G. (2021b, November 11). *Ultimate Guide to Pros and Cons of Blockchain*. 101 Blockchains. https://101blockchains.com/pros-and-cons-of-blockchain/

Mehta, M. (2020, December 4). *Hash Function in Cryptography: How Does It Work?* InfoSec Insights. https://sectigostore.com/blog/hash-function-in-cryptography-how-does-it-work/

Miller, C. (2021, October 28). *Stuart Haber and Scott Stornetta: How our timestamping mechanism was used in Bitcoin*. CoinGeek. https://coingeek.com/stuart-haber-and-scott-stornetta-how-our-timestamping-mechanism-was-used-in-bitcoin-video/

Paul, D. (2020, March 30). *The Applications of Blockchain Technology Beyond Cryptocurrency*. Digit. https://www.digit.fyi/the-applications-of-blockchain-technology-beyond-cryptocurrency/

Phillips, J. (2014, April 28). *How bitcoin really changed the world*. CNBC. https://www.cnbc.com/2014/03/28/how-bitcoin-really-changed-the-world.html

Proof of Work (PoW). (2021, July 22). Investopedia. https://www.investopedia.com/terms/p/proof-work.asp

Security, S. (2021, January 26). *Decoded: Examples of How Hashing Algorithms Work*. Savvy Security. https://cheapsslsecurity.com/blog/decoded-examples-of-how-hashing-algorithms-work/#:%7E:text=A%20hash%20function%20is%20a,fixed%20length%20%E2%80%93%20the%20hash%20value

The Ultra Blockchain Network is Built for The Video Game Industry. (2021, January 31). EOSIO. https://eos.io/news/ultra-blockchain/

What are smart contracts on blockchain? | IBM. (2022). Https://Www.Ibm.Com/Topics/Smart-Contracts. https://www.ibm.com/topics/smart-contracts

Wikipedia contributors. (2022, January 5). *Hash function*. Wikipedia. https://en.wikipedia.org/wiki/Hash_function

Yano, M., Dai, C., Masuda, K., & Kishimoto, Y. (2020). *Blockchain and Crypto Currency: Building a High Quality Marketplace for Crypto Data (Economics, Law, and Institutions in Asia Pacific)* (1st ed. 2020 ed.). Springer.