Bitcoin: Explained

Mosi Methula

First version: July 14, 2023

Last version: November 21, 2023

My father always told me to live a life of self-sufficiency. "*Own businesses; own land, farms; own cattle; produce your own electricity*," he would say. I think that's the reason for my obsession with accumulating bitcoins…

Author Note

This paper was written for fun in my spare time. I took to it my knowledge of the Peer-to-Peer (P2P) financial system we know as Bitcoin and laid it out in what I believe to be a concise and understandable way.[1] The paper can be limited in some details to be aimed towards anyone who does not yet understand a few things: what is Bitcoin? How does it work? And what benefits does it offer? It may also be read as another (God-willing) interesting take on the subject to those familiar with Bitcoin and its workings. This is all my work. However, I have provided references at the end. The paper is neither investment nor financial advice, but rather a side project on an innovation that I admire… enjoy.

[1] *Peer-to-Peer* implies without a middleman.

Contents

1. **Bitcoin: The Beginnings…….…..... 4**
2. **Bitcoin: Introduction…….…...5**
3. **Bitcoin: What Is It……..……..……..……..7**
4. **Cryptography: Introduction……...…10**
5. **Bitcoin: How Does It Work……...……22**
6. **Bitcoin: Discussion……..……....38**
7. **Author Abstract……..……..……45**
8. **Appendix……46**

1. Bitcoin: The Beginnings

The creator or creators of Bitcoin who goes by the pseudonym of Satoshi Nakamoto, published on October 31st, 2008, a white paper on the Cryptography Mailing list at metzdowd.com titled, "*Bitcoin: a Peer-to-peer Electronic cash system"*.[2] It was amidst the 2008-2009 global financial crisis objectively caused by the collapse of the U.S. housing market, due to mass defaults on subprime mortgages.[3] Banks were being bailed out by government financial packages all around the world and government money funded by taxpayers, flooded capital markets to prevent further collapse of asset prices. However, these measures could not prevent the $2 trillion loss in the global economy in 2008 alone. Satoshi published the workings of a financial system he had been creating a year prior, explaining it would allow users to send money without trusting in a third party with a special hold on the network such as a bank or repository (which will be discussed later). Finally, to fulfill this promise without compromising or collapsing the entire system, Bitcoin would utilize users' computing power and effective cryptography.[4]

[2] *Pseudonym* refers to a non-real or fake name–think a *username*. To this day, the creator of Bitcoin is unidentified.
[3] Mortgages taken on by consumers with a bad credit history.
[4] This will be discussed in further detail, but it refers to the practice of protecting data in the cyberspace.

2. Bitcoin: Introduction

To understand what Bitcoin is, I pose to you a thought experiment. Imagine a fictitious village where people live, they grow crops and keep livestock to feed themselves. In this village, however, the means of paying back any form of debt is by using a special stone, let us call it the Rock for now (just like how we use the Rand). Assume these Rocks are limited in supply, large and heavy, so the villagers instead communicate their transfer as opposed to carrying any Rocks. The only debt outstanding on a particular week is Thato who owes Lindiwe 5 Rocks. To repay Lindiwe, Thato would simply communicate the repayment directly to Lindiwe and later announce in the presence of Lindiwe and the whole village at their daily gathering that she paid her back the 5 Rocks that she owed, which Lindiwe will admit to in front of the village. Every adult in the village would acknowledge this and remember the transaction by each keeping their own *log* of Lindiwe's new ownership of the 5 Rocks.[5]

The following week, Lindiwe pays back another villager, Nandi and later at the gathering Lindiwe announces that she repaid her debt to Nandi, who now owns the 5 Rocks that Lindiwe once owned. The whole village will agree, as everyone sees the transaction in their logs from the previous week, i.e., Lindiwe was once paid those *same* 5 Rocks by Thato. What is happening here? Each transaction to repay outstanding Rocks was paid directly from Thato to Lindiwe and then directly to Nandi. The catch here is that no single person or *authority* was trusted and responsible for keeping track of the payments and thus the Rock ownership. Instead, this responsibility was *shared* amongst the community, such that they were all responsible for keeping track of who owned however many Rocks.

This means that you only own the number of Rocks that everybody agrees on or votes that you own, provable by the logs or ledgers that everybody holds. Should Lindiwe, who was initially paid the 5

[5]A *log* is a record of information–in this case transactions.

Rocks by Thato, announce at the gathering that she has paid 6 Rocks to Nandi, the villagers would dispute this, as she only owned the 5 Rocks which she was previously paid. The repayment of 6 Rocks would thus be rejected by *consensus* or agreement. The Rocks are analogous to a unit of Bitcoin; the villagers are analogous to Bitcoin users (or nodes which are basically users' computers) and the logs are analogous to the distributed log or ledger that Bitcoin users hold all around the globe, to document all transactions conducted in Bitcoin. This ledger is called the *blockchain* for reasons that will become clear in the future.

Suppose now, instead of each village member keeping track of the transaction in their native Rocks, there is now a self-appointed village chief who is responsible for documenting these repayments. The sequence of transactions remains the same: Thato repays Lindiwe, but now, she must first travel to the chief's homestead and announce her repayment to him. The chief would then open his logs and check whether Thato owns the Rocks that she wishes to repay. Lindiwe must later go to the chief stead and confirm her new ownership of the Rocks. The chief, however, might feel entitled to a portion of the debt, so he orders 1/5 of a Rock to be paid on top of the 5 Rocks as compensation for his work in recording the transactions. 5.20 Rocks are now in the 'virtual' possession of the chief, but 5 go to Lindiwe. This transaction is now more expensive to Thato than before (0.20 Rocks more). Secondly, the transaction was slower, as Thato had to first travel to the chief's homestead and later Lindiwe had to make the same trip, as opposed to them previously transacting directly. Better yet, what is stopping the chief from stealing innocent villagers' Rocks, or spitefully deflating the number of Rocks paid by a villager? The answer is nothing, as each villager puts their *trust* in a central acting authority in the handling of their monies. This chief is analogous to a bank or any other financially trusted middleman.

3. Bitcoin: What Is It

With all this in mind, let us come up with a formal definition for Bitcoin that goes as follows: Bitcoin is digital property transferable without a trusted middleman whereby all transactions in bitcoins are recorded in an immutable spreadsheet/log/ledger shared amongst users in the Bitcoin network.

Bitcoin was created as decentralized digital money, to operate outside the control of mostly banks. Much like physical paper cash, transacting in it does not require a middleman. It is simply me paying you directly in paper cash without asking for a bank's permission. Similarly, it is near impossible for one to trace back all cash payments in the hopes of identifying the previous owners of the cash–provided that the cash has been exchanged several times. By this, we can conclude that cash payments are Peer-to-Peer (P2P) or 'man-to-man', not requiring any relationship with a bank for the cash transaction to be conducted and lastly, they are relatively anonymous–e.g., down the line, I can't be identified as the previous owner of any given R10 note I once paid with.

These were the very privileges of cash that Satoshi Nakamoto hoped to replicate with his creation of Bitcoin, where transactions do not require permission from a third party, and finding the real identities of Bitcoin users and the intention of their transactions would be very difficult.

My hopes are that your knowledge of Bitcoin has improved, but there are still many concepts that I will go on to explain right after I pose to you another thought experiment. Suppose now that Thato, Lindiwe, and Nandi no longer reside in the fictitious village, but instead, are residents of the Johannesburg Metropolitan Municipality. Thato owns 3 bitcoins and wishes to send or convey them to Lindiwe as a birthday present. We must ask ourselves how it is possible to know whether Thato owns these bitcoins. Well, the simple answer is to wait for Lindiwe's wallet to show a balance of plus 3 bitcoins or just check Thato's wallet balance. However, the question above is deeper; that is, how do we in fact know where Thato got all her bitcoins from? This is important because, with this

information, we can determine her balance and assess whether she can subsequently send the 3 bitcoins. The conventional answer to the question is to have a central authority keep track of all Bitcoin transactions including Thato's, but according to Satoshi, there is no central authority such as, for instance, the Bitcoin Bank, which would be able to confirm Thato's ownership of the 3 bitcoins that she wishes to send. With a normal bank transaction, I can only send the money that is in my account and the bank stands witness to all the money coming in and out, in order to determine my final balance. Without a third party like a bank, we currently have no way to prove that Thato owns the 3 bitcoins in her 'account'.

Now, just assume with divine intervention we have managed to prove Thato's ownership of the 3 bitcoins, allowing the transaction to go on: Thato -> 3 bitcoins -> Lindiwe, but just 5 seconds after sending these bitcoins, Thato sends the same bitcoins to Nandi. This is known as double-spending, as Thato attempts to send/convey the same bitcoins to two different entities at the same time. This is very possible because bitcoins are inherently digital information, and it is very easy to make multiple copies of any type of digital information (think of pirated movies or copying and pasting words from the internet). So now, how do we determine who gets these bitcoins? Good hearts might suggest Lindiwe should, as it is her birthday and she was the first intended recipient, but the problem remains: without an entity like a bank to single-handedly witness transactions and settle payment disputes like double spending, we have no way of proving Thato's Bitcoin ownership and thus allocating the bitcoins to Lindiwe's wallet.

On the brighter side, there is a solution to both problems without the need for a trusted authority. Let us think back to our fictitious village example: for Lindiwe to claim that she owned 5 Rocks, most of the villagers at the gathering had to agree amongst each other that they recall that Lindiwe was paid those 5 Rocks by Thato. This same level of agreement or (better said) consensus is needed in the Bitcoin network. User's Computers or nodes (this word will be referred to a lot in the coming chapters) need to achieve consensus on who owns what bitcoins, how many bitcoins, and who gets

the bitcoins in the case of a user double-spending because no single node must have sole authority to settle this dispute. This gives Bitcoin its reputation for being decentralized or democratized, making you your own bank. That is because you can document and thus stand witness to the transactions in the network just like a bank would on its own.

The innovation behind Bitcoin is not the idea of an internet currency outside the control of banks or a single authority because hundreds of developers have thought this, tried this, and failed at it. The real innovation is how Satoshi was able to achieve this, by bundling cryptographic technologies and creating incentives to protect the network and allow Bitcoin users to reach consensus. These all come together to make Bitcoin operatable outside the banks' control. Rest assured; you are well on your way to understanding this.

4. Cryptography: Introduction

The topics discussed in this chapter may seem abstract or unrelated to Bitcoin, but this is not true. Think of cryptography to Bitcoin as what mathematics is to physics. These topics serve as very important foundations for Bitcoin's infrastructure. I highly recommend you do not skip over them. Should I unwillingly lose your understanding, well then, YouTube might be a great source to consult.

Cryptography is the study and practice of securing data in a network using mathematical codes and algorithms. Think about the last time you phoned your best friend; your sim card used cryptography to ensure nobody with a simple old-fashioned radio could hear the call. Also consider all the WhatsApp messages you text your peers daily, even those texts use cryptography, such that nobody else can see the text except for you and your peers. From making purchases online to connecting to your Bluetooth headphones, cryptography is used in some way. And as I mentioned above, Bitcoin uses plenty of it.

Confidentiality

To begin with, a goal of cryptography is to ensure that knowledge of certain information that is sent out through a medium like the internet or another cyberspace network, remains a secret between the sender and the intended receiver (confidential), like WhatsApp messages or online banking details.[6]

So, to keep a message private between, for instance, Thato and Lindiwe, they need a way to ensure that nobody else can make out what their message says. Notice that I never said that they need a way to *prevent* the message from getting into the hands of others. I will make this clear shortly. Thato wishes to send a confidential message to Lindiwe through a messenger saying, "Hello Lindiwe." Assume this was in the early 18th century and Thato and Lindiwe stay many kilometers

[6] *Cyberspace* is the general term for computer networks.

apart, so a messenger is necessary. Upon delivery, the messenger could simply open the letter and read the message. Therefore, the knowledge of what the message says has failed to remain a secret (confidential) between the two parties. This is a problem because in the future, very sensitive messages may need to be delivered and this messenger could *intercept* these messages, report them, or something worse.

We need to find a way to keep this message confidential and thankfully, there is a way. We could secure the message by using what is called a *cipher algorithm* (the output of a cipher algorithm). It sounds fancy, but it is simply a method used to turn plaintext into scrambled text called *ciphertext* to make it unreadable for unintended receivers.

Plaintext would be, "Hello", but ciphertext could look something like, "37hdj".
Ciphertext must be able to be turned back into plaintext by the sender and receiver, which is great because the message should stay confidential between Thato and Lindiwe. To turn the plaintext into ciphertext, it must be an *input* to a cipher algorithm. An algorithm is simply a bunch of rules and operations that take a piece of data and turn it into a final output, like the functions in math that many hate. Let us take the linear function $g(x) = 3x - 1$, where x acts as the data that we feed into the algorithm. The rules are whatever x is, I will multiply it by 3 and subtract that product by 1 to get the final output. Now, let us use a cipher algorithm called the Atbash Cipher to confuse the messenger. The rule of this algorithm is that the whole alphabet is reversed to get the ciphertext, such that A in plaintext = Z in ciphertext and Z in plaintext = A in ciphertext, making B = Y, H = S, and so forth. Note here that not all ciphertext will contain numbers.

- Plaintext: Hello Lindiwe.
- Ciphertext: Svool Ormwrdv.

"Aha!", Thato is probably thinking. I'm sure immediately, the messenger would be shocked at Thato's poor grammar, but that was the whole goal, right? Ladies and gentlemen, that is what we call an *encrypted* message, one that is encoded for it to be only readable by the sender and receiver.

For Thato to read the message in plain English, she must decrypt/decode or reverse engineer the cipher algorithm to find the plaintext. You may not have known this, but the algorithm can be guessed by the messenger sooner rather than later because it is a well-known algorithm. The first clue is that there is a repeat of two of the same letters which will both share the same plaintext letter. So, with knowledge of the atbash cipher, he might guess the 'o's' correspondence to 'l', making the remaining ciphertext look like "S v l l l." If he uses the same logic for 'S', he will get 'H' leaving him with: "H v l l l", then "H e l l l". From there the decrypting is straightforward.

This *encryption* was successful to an extent but was able to be decrypted with *knowledge* of the Atbash Cipher algorithm. It is for this reason alone that algorithms in cryptography try to accomplish some form of randomness. Think about it, if there was a way to jumble the letters in a truly random manner, which has no set pattern, it would be a waste of time for the messenger to try to decrypt. If Thato assigned H to Y, A to T, or Z to M, and so on, as well as made her message to Lindiwe a lot longer, this would make the decryption process very difficult. This type of algorithm where you personally substitute plaintext letters for random letters in the alphabet is called the *Simple Substitution Cipher algorithm*.[7] It might be important to note that skipping greetings would be helpful because it would give the messenger or any other potential interceptor fewer clues for cracking the code.

Secondly, and more importantly, the only way in which Lindiwe can decrypt Thato's message after creating her own algorithm, is if Lindiwe has the *key* to the algorithm. The key is any special piece of information, which allows a message to be *encrypted* and *decrypted*. With the atbash cipher, the key was that 'each letter in plaintext represents the letter that it would be in its position if the alphabet as we know it was reversed'. A = Z, B = Y, C = X, and so on. Other algorithms such as the *Caeser Shift Cipher* have a number as a key. To get the ciphertext, the plaintext letters must move the 'key'

[7] The probability of decrypting ciphertext which used a simple substitution algorithm by brute force is 1 / (26!), which a standard laptop could do relatively quickly.

number of positions down the alphabet. A key of 5 would turn the plaintext letter 'A' to 'F' in ciphertext or a 'B' in plaintext to a 'G' in ciphertext. Since the same key is necessary to be used by Thato and Lindiwe, it is known as a *symmetric encryption key***.** Symmetric for the same key is used to encrypt and decrypt, encryption for turning plaintext into ciphertext, and key for special information that must be known by the sender and receiver. We thus conclude that cryptography makes information confidential through *encryption*.

Asymmetric encryption

Now let us consider a problem as we look at another type of encryption. How is Lindiwe supposed to know what the key is because she cannot just guess it? An answer may be having Thato send the key first before the actual letter. The letter could say, "Hello Lindiwe, this is the key…" However, this is not effective, as the messenger could simply memorize this; wait for the ciphertext, and boom! He cracks the code using the recklessly sent key. This is known as the *key distribution problem* and thankfully, there is a way to solve this: enter *padlocks*.

You are probably familiar with padlocks, if not then I recommend a quick Google search. Padlocks have two special characteristics:

- They can be *locked* by anyone.
- They can only be *unlocked* by the padlock's owner or keyholder.

Let us implement this knowledge in our scenario above. Since Thato has already prepared the message and knows the message that she wants to deliver to Lindiwe, she could request a padlock from Lindiwe. Remember that anyone can lock the padlock onto something including the non-owner of the padlock. So, to keep the letter confidential, Thato could use the padlock delivered to her by Lindiwe and put her letter titled, "Greetings Lindiwe" in a suitcase, followed by her locking it with the padlock. This suitcase can now only be unlocked by Lindiwe or any other holder of the padlock key.

This method is quite satisfactory because we have a means of securing data with the sender and receiver being the only ones able to read the data. Let us think about this in terms of encryption, where this type of encryption also has two different keys. The padlock from Lindiwe acted as the first key because knowledge of the letter's contents is made potentially unknown to everyone but herself and Thato, by locking the suitcase using the padlock. Subsequently, Lindiwe has the second key which is to *unlock* the suitcase, allowing the letter to be known to her. The padlock is known as a public key because it can be locked by anyone, but the key that the padlock owner (Lindiwe) holds is called a private key because she is the only one who can unlock it. So, it becomes clear that these two keys are not the same, as the key used to encrypt the letter or lock the suitcase is not the same key used to decrypt the letter or unlock the suitcase. This is known as *asymmetric encryption***.** Asymmetric for the different keys used and encryption for the scrambled plaintext.[8] It is important to note that when real digital data is being encrypted with a 'digital padlock' or public key, it is turned from plaintext to ciphertext. The suitcase from the scenario has no real-world analogy but was used rather for explanation purposes–in case you wondered why the plaintext in the letter was not turned into ciphertext.

Now, one might contend that the messenger could replace the padlock coming from Lindiwe with his own, allowing him to unlock the suitcase once it is delivered with his own key. This is a very valid point against the security of asymmetric encryption, but there are real-life solutions that are out of the scope of this paper. Just note that asymmetric encryption is highly secure. Most websites on the internet make use of asymmetric/public key encryption, like when you send personal information (names, addresses, etc.). The website creates its own private key and then derives a public key from the private key. You, being the sender of information on the website, receive the public key and it is used to encrypt your personal data so that no hackers can see the data you send. The website will receive the data and use their private key to decrypt it, allowing them to see and use it as they need.

[8] Also known as *public key encryption.*

Data integrity

As I mentioned above, cryptography has the sole purpose of securing data within a network. This could be a physical network like we saw with Thato and Lindiwe, or it could be a digital network like the internet, where website visitors send encrypted data to the website servers.[9] Cryptography uses encryption to keep data confidential, but this isn't enough to keep it secure. Let us explore why. Thato might have a file with a message that she wishes to transfer to Lindiwe. As the file is sent via email, a hacker intercepts the email and changes the message. Assume that the message is not encrypted and reads, "Meet me at 8 am.", but gets changed by the attacker to "MJDJdwTD AwA 1289 am." Lindiwe will subsequently have no understanding of this message and the meeting may be missed. Cryptographers needed a way to determine whether data has maintained its *integrity* (not been changed). Clearly, the intended message above did not maintain its integrity because it had been successfully changed. It is possible to use different methods to ensure that the message *cannot* be changed, but to keep this section on par with Bitcoin, I will go on to explain a data integrity mechanism to *notify* us of undesirable changes.

Hash functions, also known as the 'Swiss Army Knife' of cryptography because they have many uses, are incredible data integrity mechanisms.[10] These are one-way algorithms/functions that compress any amount of data (messages, numbers, files, etc.) into an output that contains 64 characters of random letters and numbers, called a hash or digest. There are different hash functions, but Bitcoin uses the SHA-256 and the RIPEMD-160. Hash functions have four characteristics:

- Finding the hash of any message is simple.
- Finding the input message from the hash is impossible–this is what one-way means.
- Finding the same message with different hashes is impossible.

[9] Website servers or *webservers* for short, are computer systems that host websites, by storing and delivering them to users' computers as they visit websites.

[10] They are used to protect passwords and are important for digital signatures, as well as linking documents or data together–we will see how this is imperative to Bitcoin.

- *Changing the message will always result in a different hash.*

Using the SHA-256 hash function, the message within the file, “Meet me at 8 am.” (apostrophes excluded), produces a hash/digest of:

aff4469b51562274030eea97e93420ef8cb8289f013871df0973cfe31dacd737.

The hash of “Meet me at 8 pm.”, is:

ac20fb86f0af574326e315c7724635e75f207ed721b9c97d4401777f2fd4acd6.

As you can see, the only change in the message was [am -> pm], but it produced a random change in the hashes. There is no recognizable pattern in the hashes, which makes the hash function impossible to guess and a very important piece to Bitcoin’s infrastructure.

Let us see how the hash function provides data integrity. Thato wants to send Lindiwe a digital file. She can compute the hash of this file (remember that hash functions compress many types of data) and then send the file with its corresponding hash to Lindiwe. All that is left for Lindiwe is to check whether the file has been changed by computing the hash of the file. If it matches the hash that was sent to her, fantastic! The file has not been changed. If the file was changed by an attacker, then the hash that Lindiwe computes would not match the one that Thato sent. This is still good because they have a means of determining if their message has changed. Unfortunately, attackers are usually smarter than this: they would instead modify the file upon transmission and then compute the hash of the modified file. So now a modified file is sent along with its corresponding hash. When Lindiwe gets the unknowingly modified file and computes its hash, she will find that it matches the hash sent to her by who she believes was Thato. The hash function thus failed as an integrity check because the file was modified by the attacker without detection. We cannot have that.

So, there is a way to ensure that the attacker’s efforts to modify Thato’s files do not go unnoticed and that is to ensure that he is never able to compute the correct hash. Thato could first prepare the file and compute its hash. She could then communicate the hash by a means that the attacker cannot manipulate, such as via phone. She phones Lindiwe and has her write down the hash. The file is then emailed to Lindiwe without the hash. Upon receipt, Lindiwe would recompute the hash of the

file and see whether it matches the hash that she wrote down. A matching hash would mean that the file is not modified, but different hashes and the file is modified. The changes will not go unnoticed, which is what we want.

However, it may not always be efficient or possible to have a separate means of keeping the integrity of the data, like phone calls or texts, etc. So, this calls for (no pun intended) another means of ensuring that an attacker can never get the right hash. Recall above from symmetric encryption that we have two keys shared only by the sender and receiver. We could use the same mechanism not necessarily to encrypt the file because it is not a secret, but rather to include it in the process to find the hash of the file. Let me explain. Thato and Lindiwe can agree on a secret key (a bunch of numbers and letters for this scenario). Thato would input the file and add the agreed key to the hash function, which would give her a hash. Then she would send the file to Lindiwe with the hash but without the key**.** Now, Lindiwe would input the file and add the key to find the hash. If the hashes match, then the file is unaltered, but if they do not, then the attack on the file is detected.

These are known as *keyed-hash* functions or hash message authenticator codes (HMAC) because they use a key in the hashing process. Hash functions serve as incredible integrity checks (data integrity mechanism) because they are computed directly from the data and stay the same if the underlying data does not change.[11]

Data origin authentication and nonrepudiation

[11] Use this link to create your *own* hashes, https://xorbin.com/tools/sha256-hash-calculator.

Above in the example of bypassing an attacker's attempt to modify Thato's file without getting noticed, we needed a means to ensure that the attacker could never get the right hash. One way was communicating the hash via phone call, where the hacker could not hear the hash and secondly, we could use a keyed-hash function where an agreed secret key was added as an input into the hash function. This ensures that the only way that the integrity check can be breached without detection is if the attacker has access to the key, which he does not.

The keyed-hash function or HMAC not only operates as an integrity check, but it also points out where the file originated from. This is because possession of the key either means that Thato or Lindiwe (the holders of the key) created the file, as the hash of the original (unmodified) file had the key included in the hash function input. In this case, Lindiwe knows there is only one other person who has the key, and she was the one who sent the file–Thato. This is known as *data origin authentication*, where knowledge of where data originated from is known, which is another aim of cryptography.

There is a slight problem though. Between the two, it is known that the data or file originated from Thato, but what if there is a dispute over who the file originated from, say, because the file has incriminating information; neither Thato nor Lindiwe would want to be the known sender of the data/file. For the sake of explanation, assume the file was not transmitted through a means like email where senders are clearly shown, but instead, a shadow network where there are not. The dispute is possible because both Thato and Lindiwe hold the same key used to create the same hash. We know that Thato sent the file and Lindiwe does too, but how would a third party know this, like a lawyer or a forensics expert?

This requires *nonrepudiation* which is a means of linking an integrity check like the hash back to its original source, such that the source cannot deny creating the integrity check. Nonrepudiation will require a cryptographic mechanism to allow anybody (third parties) to verify who the creator of an integrity check is. Remember that the dispute arises because they hold the same key used to create

the hash. What if they did not? What if one key had a different purpose to the other–one to create and the other to show who created it? What we want are basically asymmetric capacities that can be brought about with different keys.

Digital signatures

If you paid attention to the encryption section of this chapter, you would know where I am trying to get at–asymmetric encryption. Excellent answer if you thought so! Although, the answer is neither right nor wrong. It is not wrong because we do not want Thato and Lindiwe to have the same keys and with asymmetric encryption there are two different keys. However, it is neither right because we will be using asymmetric encryption practically in reverse, where the roles of the private and public keys are swapped. This means the private key will be used to encrypt, and the public key is now used to decrypt the data.[12]

It would be great if Thato could just simply sign the file, she wants to send to Lindiwe, thereby binding her signature as the file's creator. This would be known as a *digital signature,* and it would verify the data's creator because that is what signatures do: they link data to their source. This way, any third party can verify the origin of the data and the Thato/Lindiwe dispute would be resolved. Let us see how digital signatures can be created. The sender (Thato) creates a digital signature by encrypting/signing the data using the private key. She would then send the data and the digital signature to the receiver (Lindiwe). She then decrypts/verifies the signature by using the public key–see it is asymmetric encryption all over again, just in reverse. The public key is known to anyone and so any third party could verify the signature, thus nonrepudiation is achieved.

However, this is not a typical method, since it is the data that is signed/encrypted making the digital signature bigger than the data! A physical signature is small and sits at the bottom of the page, so we need something similar for digital signatures. Gladly, you all have learnt about *hash*

[12] Head back up to page 14 to refresh your memory on asymmetric encryption if need be.

functions. These are glorious algorithms that can compress any amount of data into a fixed output. So instead of the sender signing/encrypting the whole data which could be a 1000-page contract, she could sign only the hash of the data which is significantly smaller.

Now let us see how digital signatures are actually created: the sender computes the hash of the data she wishes to send and then encrypts the data with her private key, which is now the digital signature. She then sends the data along with the digital signature and her public key (think how a contract could contain a signature and address) to the receiver. The receiver can then verify the signature by computing the hash of the original data. Then she would take the digital signature and decrypt it using the public key that was sent to her. Time for comparison:
If she finds that the first hash, she computes matches the second hash (the decrypted digital signature), then she can be sure the data was unaltered and if it was, the hashes will not match, thus accomplishing data integrity.

Secondly, the receiver can be sure even with modified/hacked data that the sender of the public key is the creator of the digital signature because the sender is the only one who has the private key to sign the data–remember that the owner of a private key is also the owner of the public key. So, there is data origin authentication**.**

Lastly, digital signatures also accomplish nonrepudiation**,** as the sender cannot deny creating the data. This is because the public key can be used not only by the receiver of data but also by any third party to validate the signature, as the public key is made *public* to anyone–it is not a secret, unlike the private key. This is what makes digital signatures more useful than the old HMACs above because if Thato digitally signed a file, she has no way to deny doing so. It is also worth noting that the method used to verify a digital signature never exposes the private key itself. It is just to prove the link of the private to the public key.

So, to recap, cryptography is the practice of protecting data within a cyberspace network. Cryptography achieves data protection by keeping it *confidential* through symmetric or asymmetric encryption; notifying users of changes in their data (*maintaining integrity)* by using data integrity mechanisms such as hash functions; enabling *data origin authentication* and ensuring that the sender cannot deny sending data (*nonrepudiation*) with digital signatures. *Entity authentication* is also necessary for data protection, in other words, cryptographic mechanisms that prove you are who you say you are, but this is outside the scope of this paper.

Bitcoin uses public key cryptography (public and private keys). You will see how users sign off transactions in bitcoins with their private keys, also careful readers might notice that the survival of Bitcoin is quite dependent on hash functions…

5. Bitcoin: How Does It Work

Wallets

We all know how one might hold their essentials such as their cash, credit/debit cards, ID, driver's license, and so on in their leather wallets, but there is similar software that exists for holding, receiving, and sending bitcoins. Enter *Bitcoin wallets***.** These are software files that generate and store three vital pieces of information that every Bitcoin user holds:

- *Private keys.*
- *Public keys.*
- *Bitcoin addresses.*

As you can see, you are familiar with two of the three things above and will continue to see more familiar concepts as you read on.

Private keys

Private keys are a string of 64 numbers and letters (alphanumeric characters) which are designed to function as a password to anything, but in this case to your Bitcoin wallet.[13] Think of it as the key to all your bitcoins which your possession of it allows you to spend these bitcoins. So, by inference, this private key must be known by you and only you, i.e., it must be stored securely. Anybody with possession of your private key has the same access to your Bitcoin wallet as you do. Private keys, if lost cannot be recovered because Bitcoin does not have a central authority to keep track of any private key, as everyone is their *own* bank. There are various ways in which private keys are generated, but each method must use a source of randomness to create the key. A source of randomness simply means using a method in which the inputs are unpredictable. This is necessary because a source of randomness makes it difficult for the same output or in this case key to be *regenerated*. If a key cannot be regenerated, this would make it very difficult for attackers to gain access to your wallet, making private keys secure. For instance, if you crunch your fist and rub it

[13] This is a 64-byte key which is equal to 256 bits.

rapidly across your keyboard, the output you would see on your screen would be impossible to recreate upon doing the same exercise again. So, the source of randomness was rubbing your fist across your key. This degree of unpredictable randomness is referred to in cryptography as *entropy*. The method I used has decent entropy, but not good because at any point during my keyboard swiping, there are roughly 4 to 5 surrounding keys that are likely to be picked as my fist moves in one direction. Whilst obtaining the same output is nearly impossible, there is a slightly recognizable pattern which is what cryptography goes against. See below that every subsequent key is never too far left or right as the previous one on the keyboard.

My first output: fegtyujikuytgrfedfgyuioujytgrdsjhmgfdghyjukilkjmnb.
Second attempt: dewgtuop[‘-0;pijt54rfqegtyuiop;lmnjktrewfsdfghjyu656terdf.

When generating a private key for a Bitcoin wallet, the above output would be used as a random input to several algorithms including the hash function SHA-256 to get a 64-character long key, like this one I generated from an online private key simulator: 8b256afb33ca5b85fe969e9da45a2a3009258187defc47302e3151d4136f2c4a. (*Do not* use any Bitcoin credentials mentioned in this paper, as they are just for explanation purposes.)

You could get a cryptocurrency exchange platform to do it such as Coinbase.[14] These platforms will manage your keys on your behalf, but they have the same access to your wallet as you do, which can pose a threat to your funds. These are known as *custodial wallets*. On the other hand, you can also download encrypted wallet software which would create your key from random words (referred to as seed phrases). However, you would be solely responsible for managing your own keys, giving the name *self-custody wallets*. More importantly, you could generate them yourself. This is another reason why Bitcoin is considered a decentralized financial system. You do not have to go through any

[14] This will be discussed in a little more detail in chapter 6, but they are alternative internet currencies which use similar technology to Bitcoin.

entity to create or register your credentials for your Bitcoin wallet. You can just generate your private key and everything else, then ask people to send bitcoins to your created address! Lastly, what makes private keys secure apart from a source of high entropy, is that they are ridiculously large numbers. So, it would take longer than one's lifetime to guess the right private key, i.e., it is a waste of time to do so. The largest number that a private key can be is 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936.

I will not waste my time writing it out for you, but this is a large number that can be approximated to 1.15×10^{77}. Our private key number is 62937597651984373386358880989758809001485920832937347744813879490709967285322 or 6.29×10^{75}.

Decimal and hexadecimal number systems

You may be puzzled as to how a private key such as 8b256afb33ca5b85fe969e9da45a2a3009258187defc47302e3151d4136f2c4a with letters can be converted to a decimal number. Well, this is because private keys are not inherently decimal numbers themselves. They are hexadecimal numbers. Let me briefly explain: the numbers that we encounter daily like prices at the shop or calendar dates all fall under what is called the decimal number system. This number system has 10 numbers which include 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. My current grade uses 1 and 2 to give 12 or the current price of 2-liter milk is (+-) R35 at the time of this writing.[15] On the other hand, private keys use numbers that fall under what is called the hexadecimal number system. This number system has 16 numbers! Do not worry, as you have seen the other 6 numbers before…

The numbers include 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f. The 'a' = 10 in decimal, 'b' = 11 in decimal, 'c' = 12 in decimal, 'd' = 13 in decimal, 'e' = 14 in decimal and 'f' = 15 in decimal. If you

[15] November 21, 2023

include the zero, the total numbers in this system are 16 (0 all the way to f). This may seem counterintuitive, given that some letters are numbers, but this number system is more efficient than ours (decimal). Let me show you why by using the knowledge from above. 10 in decimal equals 'a' or 'A' in hexadecimal. 13 in decimal equals 'd' or 'D' in hexadecimal. You can see that the decimal numbers were converted to hexadecimal and became a single character, thus making them more compact and taking less space or memory in a computer. More complicated conversions which I will not teach you include 100 in decimal equaling 64 in hexadecimal and 556750 in decimal (with 6 characters) giving 87ECE (with 5 characters) in hexadecimal. Again, the hexadecimal number system is more efficient than the decimal one because it can represent large decimal numbers with fewer digits.

Public keys

As the name suggests, public keys are no secret. They can and will be known by many computers/nodes within the Bitcoin network. As we saw in the Cryptography chapter, public keys were used to encrypt but were also used for decrypting/verifying digital signatures. This is its use case in Bitcoin, verifying whether someone allowed a transaction with their bitcoins to occur. An important characteristic of public keys is that they are created/derived from private keys. Therefore, private and public keys (the keypair) are mathematically connected. This makes it possible to verify that the owner of the private key that was used to spend bitcoins is also the owner of the public key that they sent for verification. This will be explained in more detail shortly, although digital signatures were discussed on page 19. The method for creating a public key is more complex. It uses a cryptographic algorithm called Elliptic Curve Cryptography (ECC) to first generate a private key and then derive a public key from it using the properties of an elliptic curve. This, however, is outside the scope of the paper.

What is important to note is that the cryptographic algorithm used to get the public key is a *one-way* algorithm. This means that it is easy to create a public key from a private key, but impossible to find

the private key from the public key. This is crucial because the public keys of Bitcoin users are exposed for everyone to see. If ECC was reversible it would be as simple as 1, 2, 3, for attackers to gain access to innocent nodes' wallets. Public keys are also very long, typically 66 to 130 characters, and are in hexadecimal format. The private key above has a corresponding public key of 0391dcda1c006f0d0d0c465567dceff77b3ec4d93590b60b81dae6877a05813023.

Bitcoin addresses

Bitcoin addresses are user identifiers in the Bitcoin network like usernames. When users send or receive payment in bitcoins, they use their Bitcoin address as their 'account name/number', which is why Bitcoin is considered relatively anonymous, as it would be very difficult for an unknown user to tie a Bitcoin address to a person's real identity. Thus, the address is in *no way* linked to their identity. They are derived from public keys and are therefore mathematically linked to both the private and the public keys. Bitcoin addresses are created by hashing the public key twice, using two different hash functions. The SHA-256 and then the RIPEMD-160.

Creating the address looks like this: Bitcoin address = RIPEMD160(SHA256(public key)). So, the corresponding Bitcoin address of the public key above is bc1qc6md7dausky4xg2gy49hx49706cwgljy9rvpxu. You can see the obvious difficulty in tying this address to me.

Transactions

The Bitcoin wallet is not only responsible for storing private and public keys, but it also enables Bitcoin users to transact with themselves. The wallet generates inputs, outputs, a transaction identifier, and a digital signature to create a transaction. To keep this section simple, I will go on to explain outputs, the transaction identifier, and the digital signature.

Let us reconsider the transaction between Thato and Lindiwe. Thato wishes to send 3 bitcoins to Lindiwe. Thato's wallet would generate a transaction consisting of the amount she would like to send Lindiwe and the Bitcoin address of Lindiwe, which are the outputs of the transaction. The transaction will also consist of the hash of all the information in the transaction, which is the transaction identifier. It would also include Thato's digital signature; her public key and the transaction fee Thato wishes to pay (we will see shortly who this fee will go to). In terms of Bitcoin notation, the transaction might look like this:

Thato's Bitcoin address: [1GUFT9ziPzEMBGHEnd6XkMZfNjLvCmcCRN]
Bitcoin amount: 3.00
Transaction fee in bitcoins: 0.0001
Lindiwe's Bitcoin address: [139zAfHSu2YKies2Pwid3Qn67JAXLAyDGw]
Thato's public key: [03f1ad69e7c91b3fc9df70bacc2f9b4eef9cab849eaec6defd77f5543330b8b4e0]
Transaction ID (hash of all the above)
[d442ae46ff9fe886fddc3df8bb62824db75aa16076b868d5b891632d453475bf]
Digital signature.

This again is a simplified explanation, nevertheless, it shows the vital components of a Bitcoin transaction. The next step is for the transaction to be verified by nodes in the Bitcoin network. The transaction would be sent out to the Bitcoin network, where nodes will take the transaction and identify the digital signature and the transaction ID. The digital signature was created by taking the hash of the transaction ID and encrypting it using the private key of the sender (Thato). The job of these nodes is to verify the transaction by ensuring that it was not altered by an attacker. To do this they would take the hash of the transaction data available to them to recreate the transaction ID. Then they would take the digital signature appended/added to the transaction and decrypt it using Thato's public key to find the hash of the original transaction data (original transaction ID). If the hash that the nodes compute matches the decrypted digital signature, then the transaction is

unaltered and thus considered valid, but if the hash does not match the decrypted digital signature, the transaction is deemed invalid. Let us see why. Recall that digital signatures are integrity mechanisms that notify parties of any changes to the data because these signatures are computed from the data itself. (It differs from handwritten signatures which always look the same.) From previous knowledge, if the message or data that is hashed changes, a new subsequent hash will be created. That is the same for Bitcoin transaction data. A hacker might want more bitcoins to be sent to his address, thus changing the recipient's Bitcoin addresses and the number of bitcoins transferred. The transaction ID which reflects the hash of the transaction data would consequently change. This would result in a mismatch with the decrypted digital signature, causing the transaction to be rejected.

Earlier, I brought up two very important questions, the first being, how could we determine a user's balance of bitcoins without a central authority. And the second, how do we resolve the double spending of bitcoins by a malicious user, again in the absence of a central authority? The answer was that Bitcoin users need a way to reach a consensus regarding the users' balances and who gets what bitcoins in the case of double spending. This consensus is brilliantly brought about by two mechanisms: the *blockchain* and *mining*, which I will discuss soon.

The first step in achieving consensus on users' Bitcoin balances is to make every transaction in Bitcoin public for everybody to see.[16] This way, each bitcoin in existence can be traced as it is received and spent by nodes, allowing consensus on how many bitcoins every Bitcoin user owns. Recall the village example in Chapter 2, where Thato would announce her Rock transaction in the presence of all the villagers allowing them to keep track of her and the recipient of the Rock's balances. The same thing will happen with Bitcoin: users who wish to spend their bitcoins will have their wallets generate a transaction and then *propagate* it throughout the Bitcoin network (or the

[16] Merkle roots, which you will see later are hashes of a block of transactions. These are instead recorded in the blockchain, as opposed to all the transactions, as they take up significantly less space.

village with respect to the Chapter 2 scenario). Propagation simply means that the user's unconfirmed transactions are broadcasted or shared with as many nodes as possible. Nodes in the network are essentially connected to each other like how users in an online game server would be. So unconfirmed transactions are spread out to some nodes, and they would broadcast the transaction to their connected peer nodes, like how a villager who did not attend the daily gathering could be told by several attendees of Thato's transaction. This creates a propagation effect. As the transaction is propagated, it is independently verified by each node. If the transaction is deemed invalid, it will not be propagated any further by a node, but if it is deemed valid, it will continue being shared.

Double spending

Let us consider a given number of bitcoins being double spent. Nandi sends the same 3 bitcoins to Thato and then Lindiwe a minute later. Two transactions would be initiated, one to Thato's Bitcoin address and the second to Lindiwe's Bitcoin address. Because of latency, due to different geographical distances and network speeds some nodes will receive either of the two double spends and reject the other. So, the Bitcoin network is presented with two conflicting transactions. The same question can be asked as before but with a more in-depth answer. How do we determine who gets the bitcoins between Thato and Lindiwe? Satoshi was aware of this problem and the need to give preference to the first intended recipient. So, he needed a way to time-stamp/process transactions such that the very first of the double-spent transaction would be more likely to go through.

He realized that the only way to do this without a central authority was to have Bitcoin nodes *compete* against each other by solving a 'computationally intensive puzzle' and the first node to solve it gets the privilege of processing transactions, with the hope that the first intended recipient of the double spend gets her transaction processed first. Remember that Bitcoin has no central authority operating the network, as this responsibility is shared by all nodes. Allowing them to compete to process transactions would (in theory) allow any node to win, which rules out the possibility of a

single node processing all the transactions, achieving decentralization. Secondly, the reason that both double spends would be considered is that they will be propagated and some of the competing nodes will get one whilst rejecting the other and vice versa. Only one node can solve the 'puzzle' and therefore process the Bitcoin transactions at that given point. Depending on which of the double-spent transactions that the node gets, either of them could be processed.

Proof-of-work mining

Proof-of-work mining is likely the most important aspect of Bitcoin. It involves the use of a specialized computer chip to do mathematical computations. Mining is essentially the competing process that I alluded to above. To do so, nodes attempt to create a *block* full of transactions to process them and the block of the winner will be added to the public ledger for all Bitcoin users to see. A new block could be thought of as a new page of transactions that will be added to the ledger. We call it proof-of-work mining because it signifies that intensive computational work is done to create the block with the hopes of getting a reward like you know–digital gold... The mining process involves a specific node called a miner collecting unprocessed transactions, verifying and cryptographically binding them. Then, attaching what is called a *block header* which stores important information about the block in creation, such as a hash summary (Merkle root) of all the transactions collected; the hash of the previously mined block in the blockchain, and lastly a random value called a *nonce*. There is one more important aspect to mining which is in fact what makes it a difficult process. That is the competing or solving of the computationally intensive. First, the Bitcoin protocol (software governing the rules) specifies the number of leading zeros that a hash must have. The first miner to compute a hash that has the same number of leading zeros or more will be the miner that solves this puzzle and gets to add his newly created block full of transactions to the public ledger, thus processing all the transactions in the block.

The reason that this task is seen as solving a puzzle is that the *nonce* is the only value that can change to find the hash below the threshold specified by the protocol. What I mean is that the inputs

into the SHA-256 hash function will be the block header information: the hash summary of all the collected transactions (Merkle root), the hash of the previous block, and the random value or nonce. The hash summary and the hash of the previous block in the blockchain do not change, but the nonce value must be increased in order to find a suitable hash with the specified number of zeros. I want you to think back to the properties of hash functions. They produce a random change if the input (nonce) changes, so there is no recognizable pattern when creating hashes. This will result in miners trying billions or even trillions of different nonces every second in the hopes of finding a lucky one that produces the hash with a certain number of leading zeros, all because of the random nature of hashes.[17] It is basically a gamble or game of luck! Once the miner finds the correct hash, he will broadcast his block to the other miners who will verify it and accept it, then the block will be propagated throughout the network. The next mining round will start after the block is confirmed by the other miners.

The reason that mining was created to be challenging is to give enough time for transactions to find their way into miners' blocks in order to be processed, as well as for previously created blocks to propagate far enough throughout the network, such that most nodes have added the new block/page to their public ledger. The last part is crucial because the consensus or agreement amongst every Bitcoin node is that the public ledger with the greatest number of blocks or pages of transactions is the *valid* one. If there is a node holding a public ledger with fewer pages of transactions, then that node will discard his ledger and adopt the longest ledger held by most Bitcoin nodes. I will revisit this in more detail, but now let us look at the difficulty level of mining. The Bitcoin protocol aims for a block creation time of ten minutes, again, to give transactions enough time to be processed, but at the same time for the transactions to be acknowledged by most of the Bitcoin network. The difficulty level is adjusted depending on how many miners are competing at a point in time. The more they are, the more hashes will be generated in the hopes of finding the correct one, which would

[17] The mining speed is usually measured in gigahashes per second (GH/s) which is a billion/s, but some miners compute trillions of hashes per second so are measured in terrahashes per second (TH/s).

theoretically result in the correct hash being found more quickly. To ensure that the mining time remains around ten minutes, the difficulty level is subsequently increased by raising the number of leading zeros of the hash specified by the Bitcoin protocol. This will increase the number of nonce values required for generation, thus increasing difficulty and allowing the mining time to stay around ten minutes. Conversely, if there are fewer miners competing at a given point, fewer hashes would be generated, which in theory would extend the solving time past ten minutes. The difficulty level is also adjusted, by reducing the number of zeros required in the correct hash, so that fewer hashes would have to be computed to find it. When considering a double-spent transaction, miners would receive either of the two transactions, not both, as the miner would reject the other on the basis that it is using the same bitcoins that another transaction that they have already collected is using. Therefore, the transaction collected by the first miner to solve the hash puzzle is effectively the transaction that will be processed, and the other double spend will be rejected by all other miners, due to there being a verified existing transaction containing the same bitcoins in the ledger.

As we saw, billions or trillions of different nonces are typically computed every second to find the correct hash which takes large amounts of hardware and computing power–costing lots of money. No miner would do this without an incentive, but thankfully Bitcoin does have one which is more bitcoins! The winning miner receives all the transaction fees of the several thousand transactions processed by him in his block–decent money, but the second reward is called the *block reward*. This block reward and the transaction fees are only awarded to the winning miner who gets to add his block/page to the public ledger. The catch here is that every single bitcoin in existence is brought about through the block reward. These bitcoins are awarded by the Bitcoin protocol, to the miner who solved the puzzle. Using this logic, we could say that the winning miner 'minted' the bitcoins. The number of bitcoins minted halves every four years. From 2008 to 2012, 50 bitcoins were minted every ten minutes, which halved to 25 from 2012 to 2016, all the way to 6.25 bitcoins which are currently being mined every ten minutes. This will last until about March 2024 before halving to 3.125 bitcoins. Finally, the maximum supply of Bitcoin will only ever reach 21 million, which is when

the halving period is expected to stop, in the year 2140! Satoshi designed Bitcoin to sort of mimic gold in terms of its supposed scarcity. If it is limited in supply, then its value will be driven up due to demand outpacing the supply of Bitcoin, hence you might have heard Bitcoin being called 'digital gold'.

The blockchain

If we revise some of the terminology used in the previous section, the public ledger becomes the *blockchain,* and the pages within will only be considered blocks from here on. The blockchain is the distributed ledger consisting of a chain of blocks of transactions, dating back to the very first transaction made. It is considered a chain because each block is cryptographically connected to the one created before it, thus establishing a linear (straight) chain network. The connection is brought about as the hash of the previous block's header is included in the header of the block after it. When blocks are added to the blockchain, they are there forever and so is their transaction history. This makes the blockchain *immutable* or unable to be altered once the blocks are added, making it easy for nodes to determine Bitcoin ownership. 'Blockchain' is quite a buzzword in financial or tech news and can be an intimidating search when confronted with jargon like decentralization or cryptography, but the blockchain if anything, is one of the simplest aspects of Bitcoin to understand. It is simply the Bitcoin transaction vault. To check who owns what and how many bitcoins, any node can make its way to the vault where all the transactions are stored and search for itself.

Here, Satoshi sums up his creation of the blockchain perfectly: *"It is a global distributed database, with additions to the database by consent of the majority…"*

Let me pivot back to hash functions. They are needed for digital signatures, mining, and linking every block in the blockchain. The reason that the blockchain is seen as a robust *integrity mechanism* is that it will notify nodes if any transactions are to be altered. How? Well, by now we know that hashes are computed from data and any change in the underlying data would produce a completely

different hash. Since every block has the hash of the previous block, an attacker who tampers with a transaction; perhaps replaces a Bitcoin address with his own or changes an amount paid, the Merkle root which is the hash of every transaction in the block would change. This would consequently change the hash of the block, differing from its original hash found by the miner. The next block which is supposed to have the hash of the changed block would no longer have it, making that block and every subsequent one invalid. This would essentially cause a break in the 'chain', making the blockchain invalid!

Proof-of-work consensus mechanism

However, Bitcoin has a defense against such hacks which is why until this day, the Bitcoin blockchain has never been hacked. This defense is called the *proof-of-work consensus mechanism*. It means that users agree that the blockchain with the most work done is the valid one. How is work done? Well, through the proof-of-work mining which I discussed extensively. Therefore, the blockchain with the greatest number of blocks mined is the one with the most work done and the proof of this is the length of the chain. Any attempt to create a new blockchain would simply fail because nodes will reject it through consensus, as it is shorter than the valid one. When an attacker tampers with a transaction and thus a block, the length of his blockchain essentially cuts to the block that he tampered with. Furthermore, to have his blockchain considered valid, i.e., successfully hack the blockchain, he would need to redo the work or re-mine every block following the tampered one to calculate the correct hashes and reform the chain. However, whilst he does this, new blocks are being mined continuously extending the length of the chain that he wishes to hack. Because mining uses extensive computing power and thus vast amounts of electricity, rewriting the blockchain would require exorbitant amounts of money to obtain 51% or more of the total mining power in the Bitcoin network because it is then, he would be able to mine faster than all the other miners combined. Unless this is done, he will never catch up to *honest* miners adding valid new blocks and thus never successfully hack the blockchain. This hacking process is known as the '51% attack' and is extremely unlikely to be successful, hence we say that the blockchain is immutable.

This is one of the reasons why providers of Bitcoin services such as payments or exchanges, typically require transactions to undergo multiple confirmations like six before a transaction is fully processed. The first confirmation happens when a transaction is included in a block. Subsequent confirmations occur when more blocks are added after the initial block, as the transaction data becomes embedded deeper in the blockchain. This increases the difficulty of reversing the transaction, as the work of mining the block and subsequent ones would need to be redone. Because of this, the transaction time could increase to several hours on average, depending on the payment provider in use.

However, the costs of hacking the Bitcoin blockchain far outweigh the benefits. The only short-term gain would be rewiring payments back to the attacker's address, but not long after the price of Bitcoin would (in theory) tank to nearly nothing. This would be the result of the hack making news headlines globally, damaging the reputation of Bitcoin's security and subsequently wiping out the value of the attacker's stolen bitcoins. This would provide no return on the huge sums of money spent to just build the necessary infrastructure. Think about it like this, the biggest miners are companies worth hundreds of millions or even billions of dollars with the operation of mining. Going up against them would be quite challenging…so, it would only make sense for malicious nodes to, instead of attempting to re-mine the blockchain, rather contribute to the mining of the longest chain. They would not only make more in the long term from block rewards, but they would also increase the underlying security of the blockchain. This is because the node would bring more computational/hashing power, increasing the difficulty of mining. Any other potential attacker would have to redo more proof-of-work making the process less feasible.

Forks

The Bitcoin blockchain is classified as linear meaning that blocks are created one at a time in a straight line to form a chain. This is true to some extent, but the reality is that the Bitcoin blockchain has been split into more than 100 different chains, creating a fork-like shape as opposed

to the linear chain I described earlier. Bitcoin runs on users' computers or nodes that run the software necessary to operate the Bitcoin network. There are several, but the most used is called Bitcoin Core. Nodes, particularly miners, may not always run the same version of Bitcoin software. Similarly, we could think of Apple iPhone owners: some run the latest version of IOS, others procrastinate the update and run slightly older versions, whilst others are limited to older versions with outdated iPhone models. Bitcoin nodes can independently propose minor changes and updates to the software such as security upgrades or new features, etc. These proposals undergo community assessments and if most miners and enough nodes support the update, then miners would begin to install it and run the latest version. These updates are called f*orks*. There are *soft* forks, *hard* forks, and *temporary* forks. Soft forks are changes to the Bitcoin software code or just protocol that nodes are not required to install. Nodes running older versions can continue propagating transactions and mining blocks, which would still be recognized by nodes running newer versions of the software. This does not cause any disruptions to the network nor any splits in the blockchain, but rather slight changes to the protocol. A soft fork is *backward compatible* with nodes running older software, meaning that users not running the latest software version would still participate in the same blockchain.

A *hard* fork is a change in the Bitcoin protocol that requires nodes to install in order to continue participating in the blockchain. These changes make the rules under previous versions invalid under the new version, which means that blocks mined by nodes running previous versions will be rejected by those running the newer software. Besides, this does not stop miners using the old software from mining, thus a split in the blockchain is created–like the shape of a fork. There would be more than one branch: one with mined blocks under the new protocol and the other with mined blocks under older protocols. These branches would continue to grow as long as there are miner nodes running the respective versions of the software. What is important to note is that alternative branches effectively create their own currencies due to the different rules that they follow. Examples of these would be

Bitcoin Cash, Bitcoin SV, and Bitcoin Gold. They each have their differences from Bitcoin and each other, but the discussion of these is out of the scope of the paper.

Temporary forks happen when miners add blocks to the blockchain at the same time (solve the hashes at the same time). Both miners would propagate their blocks and nodes would add either, not both, of the blocks to their blockchain. This would create a temporary split accompanied by two branches with either one held by a node in the network. This type of fork is resolved by going into an 'overtime'. See, after the miners found their solutions, a new round of *hashing* begins–as always. The branch that the future winning miner holds will become the valid branch, resulting in the other branch being discarded by all nodes that hold it. The fork cannot be prevented from happening, but it is resolved with ease. The block that was found by the previous miner but discarded in the new valid blockchain is called the *orphaned* block.

6. Bitcoin: Discussion

Well, by now you understand Bitcoin as a currency and as an intricate financial system. This chapter will just convey some ideas for you to play around with. Again, these ideas are not there to influence your investment in Bitcoin because I do not stand to gain anything from that.

Bitcoin is notional, meaning that it is just an idea on the internet. It is seen to have no intrinsic value because it does not produce anything, and it has no physical use cases. Stocks are seen to have intrinsic value because they are backed by companies that produce cash flows and profits; bonds have intrinsic value as they produce coupon payments; houses and gold have physical use cases and so have intrinsic value. Bitcoin, however, does not because bitcoins are inherently digital information. They do not produce cash flows or profits and they cannot be used physically, as they exist purely on the internet. It therefore makes it apparent that the value of a bitcoin cannot be measured against its intrinsic value. So where does the value of Bitcoin come from then? There is no definitive answer because, whilst price is based on market forces, value is usually the result of subjective analysis. Consequently, the value of Bitcoin would come from a combination of many factors.

Speculation is one reason for the price volatility and rises in Bitcoin prices. People simply buy Bitcoin in the hopes that its price will go up, however, another group of people wish to do the same thing. So, they buy bitcoins from the first group, but they would only be willing to sell their bitcoin at a profit and so they do. The price of Bitcoin gets pushed up higher as the process repeats itself. What I described is known as the *greater fool theory*. This is because, with Bitcoin's lack of intrinsic value, people would sometimes only buy it if they could sell at a higher price and not because it produces anything like a stock or bond would. The theory goes: "I am a *fool* for buying an overpriced asset, but I could just find a greater *fool* to sell it to at a higher price." This creates what is known as an asset

bubble, where the price of Bitcoin would eventually tank when the bubble pops.[18] History has shown this time and time again. Some value Bitcoin based on its current and forecasted adoption by everyday people and institutions, whilst others recognize the benefits that Bitcoin offers and value it accordingly. These benefits include Bitcoin's artificial scarcity, decentralized clearance mechanisms, superior security, transaction transparency, and identity masking system. Then there are more technical valuations in which people argue back the Bitcoin price or even make it undervalued. Bitcoins are minted through mining; however, this process is a result of large hardware and electricity costs. Thus, the theory goes that users who choose to mine could be seen as *buying* bitcoins into existence at these costs. Different valuation models would then factor the costs and the number of bitcoins in circulation to come up with a value. In addition, Bitcoin has a halving period every four years, meaning less Bitcoin would be minted for every hashing round. This coupled with the high, rising costs of mining would result in higher valuations for the digital asset.

The benefits and drawbacks of Bitcoin

Bitcoin, as one said is, "gold without its defects…" The reason is that gold is a store of value because it is relatively limited in supply, but so is Bitcoin. Although Bitcoin has a *predetermined* supply of 21 million; as far as gold is concerned, there are only estimates as to how much gold is left to be mined. What is interesting is that an estimated 4 million bitcoins have been lost forever due to lost private keys or sending bitcoins to non-existent addresses. This means that with fewer bitcoins in circulation, the perceived value of Bitcoin would likely rise over time, resulting in (deflation) a single unit of Bitcoin being able to buy more units of any given currency in the long term. Secondly, Bitcoin is significantly more divisible than gold, as the smallest unit of Bitcoin is 1/100 000 000 bitcoins, which is known as a 'Satoshi'. It is much smaller than a dollar cent, whereas the smallest unit of gold that an investor can typically buy is a one-gram gold bar, ranging from $74 on secondary

[18] A *tanking* price is one which drops drastically.

markets at the time of this writing.[19] Self-evidently, Bitcoin would be more suitable for smaller transactions than gold. Bitcoin is also more transferable because it is digital. It takes minimal effort to send any given address some bitcoins, but to transfer gold requires vehicles to carry it if it must be transported over long distances. It is also very accessible because setting up a Bitcoin wallet is quick and does not require any external documents or travel, except for an internet connection. Now, sending money cross-border (out of a country) has always been a notoriously slow process taking anywhere from several days to weeks, due to several intermediaries such as domestic and foreign banks, payment processors, clearinghouses, and other regulatory bodies. Bitcoin bypasses these burdens, as it is not regulated by any one government or institution, allowing it to be sent across the globe without being slowed by regulatory checks. A cross-border transaction would take roughly the same time as any other typical Bitcoin transaction which is between ten minutes and several hours, making the system fast and efficient for such transactions and remittances.[20] Transaction fees in Bitcoin are not proportional to the amount of Bitcoin spent because miners rely heavily on the block rewards as compensation for their work. The transaction fees appended to Bitcoin transactions fluctuate from time to time, depending on the network activity. However, a conservative estimate of the average fee could be between $1.20 and $3.00 or between R21.50 and R53.85 at the time of writing.[21] For high-volume transactions (big payments) these transaction fees would be a bargain because banks would charge fees variable with the transaction amount. Ultimately, their fees would greatly exceed the fees charged by miners, making Bitcoin a cheaper means of payment in that regard. The transparency in the Bitcoin network allows it to be decentralized and highly secure. As payments are made public to every user, the flows of bitcoins can be traced, allowing consensus to be reached. It also allows everyone to keep their own copy of the longest blockchain, which, as we saw, is a crucial defense against possible attacks in the network. Lastly, decentralization ensures that there is no one point of failure in the Bitcoin network. This means that if one node were to

[19] September 19, 2023.
[20] Money transferred by workers in a foreign country back to their families and relatives in their home country.
[21] September 19, 2023.

malfunction or go offline, it would not cause any damage to the blockchain because it is not the only node with a copy of the blockchain. So, the blockchain and all the transactions in it will always exist if many nodes have a copy and honest nodes hold most of the mining power. Bitcoin's anonymity is more of a "double-edged sword", as it could be used for legal and illegal transactions, but these transactions could bypass regulations that limit trade with other countries. It would be relatively easy to send bitcoins to say, a party in need of money, but still go unnoticed by either government. This has happened and continues to because Bitcoin addresses have no tie to the identities of its users.

Bitcoin, however, is not all roses and sunshine. For one, the price of Bitcoin over the years has shown lots of volatility because it is a popular speculative instrument, and they are not backed by anything but its underlying security. In theory, it would pose a threat to payments, as prices could rise and fall quickly, complicating point-of-sale (POS) procedures and subsequent currency conversions by the merchant/seller. As I mentioned above, Bitcoin transaction fees are not variable with the size of the transaction, but rather the network activity. Miners must pick around 2000 transactions to include in their blocks because they have a limited amount of space (1 megabyte/MB). Consequently, users effectively compete to have their transactions included in blocks, so the transaction fee must be attractive enough for miners. This results in high fees at times of high network activity, making Bitcoin unsuitable for microtransactions, as the fee could exceed the cost of the goods or services. Unfortunately, Satoshi hoped for Bitcoin to be used for these very payments, but as Bitcoin grew in popularity, microtransactions became unfeasible. If blocks were to be bigger, yes, they could store more transactions lowering the fee, but they would take longer to propagate requiring faster network speeds, more storage space, and thus hardware. All this would add to the costs of nodes storing the full blockchain and decrease its decentralization. There are add-ons, so to speak, which the

Blockchain can utilize to make transactions very cheap, but these are out of the scope of the paper.[22] Similarly, the blockchain is not scalable because only a single miner can process about 2000 transactions every ten minutes, which is about 5-7 transactions per second. If Bitcoin were to be adopted by the whole world as a reserve currency, so to speak, the network would be extremely congested, significantly slowing down transaction speeds, greatly inflating transaction fees, and hurting households and businesses around the globe. (I am assuming in this case that we would not make use of the different scaling solutions in the Bitcoin network.) This is known as the *scalability problem*, which has resulted in Bitcoin being hard forked with changes such as bigger block sizes and faster mining speeds to name a few. There is also concern about the damage that Bitcoin mining does to the environment due to its high electricity usage. In the long run, as miner nodes compete and become more powerful, the difficulty level of mining increases, resulting in more computations and higher electricity costs. The byproduct of mining and the cooling of the hardware (as the hardware gets hot) create carbon dioxide emissions which is a known danger to our climate. There is also the worrying use of Bitcoin by hackers, money launderers, and people buying illicit products online because of the relative anonymity that Bitcoin users have, as well as their intentions being invisible. Major governments have acted on this concern by banning Bitcoin outright, such as Bangladesh, Iraq, and Qatar. Bitcoin is in fact the primary medium of exchange used on illegal online black markets. Now, I need you to think back to the proof-of-work consensus section. Recall that reversing transactions is extremely difficult and has never been done in the blockchain because you would need to control 51% or more of the total mining power. This is unpleasant in some cases because millions of bitcoins have been lost in transit by sending bitcoins to non-existent addresses. Once bitcoins are sent to null addresses, the transaction is in the blockchain forever and they can never be recovered. Also, bitcoins can be lost by simply forgetting your private keys to your wallet.

[22] See this article about Bitcoin's solutions to high fees and slow transactions times, by JC jawe.sol at https://qjawe.medium.com/segwit-lightning-network-guide-2c5c5861a88.

Another unfortunate reality is that Bitcoin is *recentralizing,* which is the reverse of decentralization. For instance, when Bitcoin first started, anyone could mine using their laptop or more specifically their laptop's CPU and later GPU. However, mining now is competitive to the point where miners operate whole warehouses full of specialized mining hardware. This creates a huge *barrier to entry* for the mining industry, as someone wanting to start out would have to compete with people pouring hundreds of thousands or millions of dollars to create blocks, which is something that not everyone can do. So, miners have come up with a solution called mining pools, where users combine their computing resources and work together to mine blocks. If one of the miners in the pool mines a block, he would share the block reward disproportionately with the other miners, as they all collaborated to find the block. In fact, all the Bitcoin blocks are produced by 13 of the biggest mining pools in the world! Another case of recentralization is that the number of nodes storing the full Bitcoin blockchain is decreasing because of its ever-expanding size. There are two types of nodes in Bitcoin: lightweight and full nodes. Lightweight nodes are connected to the blockchain but do not store the full blockchain. These are usually what wallet nodes are and have the sole purpose of sending and receiving bitcoins. Full nodes, however, store the full blockchain in order to propagate blocks to other nodes (some miners are full nodes). Anyone can become one, but not everyone does, as it is costly to have the right hardware in place to store a continuously growing 400 gigabyte (GB) database. Secondly, there is no reward and thus little incentive for storing the blockchain, except for goodwill. Full nodes are essential for decentralization and for keeping the blockchain operational, but their decreasing numbers are quite worrying. There are other cases of recentralization when it comes to the Bitcoin software which most nodes operate and the suppliers of mining equipment, but these are not nearly as concerning.

Finally, something which I have not discussed is the term *crypto*. It is the general term used to describe Bitcoin, blockchains, and alternative internet currencies. Some differ slightly from the Bitcoin protocol, whilst others greatly, but Bitcoin was the first ever of these currencies to be created. Every single one of them makes extensive use of cryptography to secure their own transactions and

blockchains. So, it is for this reason that they are called cryptocurrencies–crypto being short for cryptographic. Examples of these alternative currencies or altcoins include Ethereum, XRP, and Litecoin. Crypto exchanges are platforms enabling cryptocurrencies to be bought, sold, and exchanged, with big names including Binance, Coinbase, and Crypto.com. Crypto has lacked explicit regulations by governments in the past. Regulatory frameworks are crucial for protecting investors, consumers, and businesses using cryptos. It is no secret that governments have rushed to tax cryptocurrencies, but protecting the industry is worth a debate. There are various reasons for the slow pace of comprehensive Government regulations for crypto. Firstly, its globalized and decentralized nature creates a lack of government consensus and coordination, as each government is different with respect to laws, culture, risk appetite, etc. Then, the complexity of crypto and the nascency of the industry forces governments to consult with industry experts and government officials before making decisions, thus taking time.[23] More commonly, skeptics of the slow process have suggested government hostility as a prime reason because they cannot directly tax the blockchain, they cannot control it, and neither can they identify everyone on it. The point is that the lack of regulation has slowed crypto's adoption and continues to worsen its reputation, as bad actors continually exploit the industry and its honest participants.

[23] *Nascent* implies a young and developing industry.

7. Author Abstract

This paper was written as a primer to Bitcoin, the blockchain and somewhat to cryptography. I asked many people in June what they understood about Bitcoin, but the answers were largely underwhelming, leading me to conclude that there is a huge misunderstanding about the system. I consulted my previous knowledge from books and two research papers to explain Bitcoin as money, as well as its operations under the hood. Bitcoin was created to be an apolitical, fully transparent, digital form of money. The key to its decentralization is the consensus it achieves by making transactions public in the blockchain. This allows nodes to determine the most valid chain, creating a globalized version of transactions. The blockchain employs cryptographic integrity mechanisms such as hash functions to effectively make itself an integrity mechanism, in order to counter fraud and hacks. Miner nodes mint bitcoins and help secure the blockchain, full nodes keep it decentralized and wallet nodes make use of the blockchain for transactions. Developers have built off the source code of Bitcoin and altered it to create a whole industry called Crypto. Nevertheless, due to Bitcoin's lack of intrinsic value and speculative nature, it is often seen as a scam. Without knowledge of the purpose of Bitcoin and how it achieves this, it is an easy conclusion to make. Albeit with a lot of effort, you now have this knowledge…

8. Appendix

References

Agashe, A. Detroja, P. Mehta, N. (2021). Bubble or Revolution: The Present and Future of Blockchain and Cryptocurrencies. Paravane Ventures.

Bashir, I. (2020). Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum and more. Packt Publishing.

Champagne, P. (2014). The Book Of Satoshi: The Collected Writings of Bitcoin Creator Satoshi Nakamoto. E53 Publishing LLC.

Chason, E-D. (2019). How Bitcoin Functions As Property Law. William & Mary Law School.

Gerrard, D. (2017). Attack of the 50 foot Blockchain: Bitcoin, Blockchain, Ethereum and Smart Contracts. CreateSpace Independent Publishing Platform.

JC jawe.sol. (2018). SegWit & Lightning Network Guide. Medium.com. https://qjawe.medium.com/segwit-lightning-network-guide-2c5c5861a88

Martin, K. (2020). Cryptography: The Key To Digital Security, How It Works, And Why It Matters. W.W Norton & Company, Inc.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Satoshi Nakamoto.

Saint Bits LLC. (2023). Bitcoin P2P e-cash paper. Bitcoin.com. *https://www.bitcoin.com/satoshi-archive/emails/cryptography/1/#selection-29.673-29.1060*

Satoshi Nakamoto Institute. (2009). Bitcoin open source implementation of P2P currency. Satoshi.nakamotoinstitute.org. https://satoshi.nakamotoinstitute.org/posts/p2pfoundation/3/

SD Bullion Inc. (2023). 1 Gram Gold Bars. Sdbullion.com. https://sdbullion.com/gold/gold-bars/1-gram-gold-bars

2007-2008 financial crisis. (2023, September 10). *In Wikipedia.* https://en.wikipedia.org/wiki/2007%E2%80%932008_financial_crisis

XORBIN.COM. (2019). SHA-256 hash calculator. Xorbin.com. https://xorbin.com/tools/sha256-hash-calculator