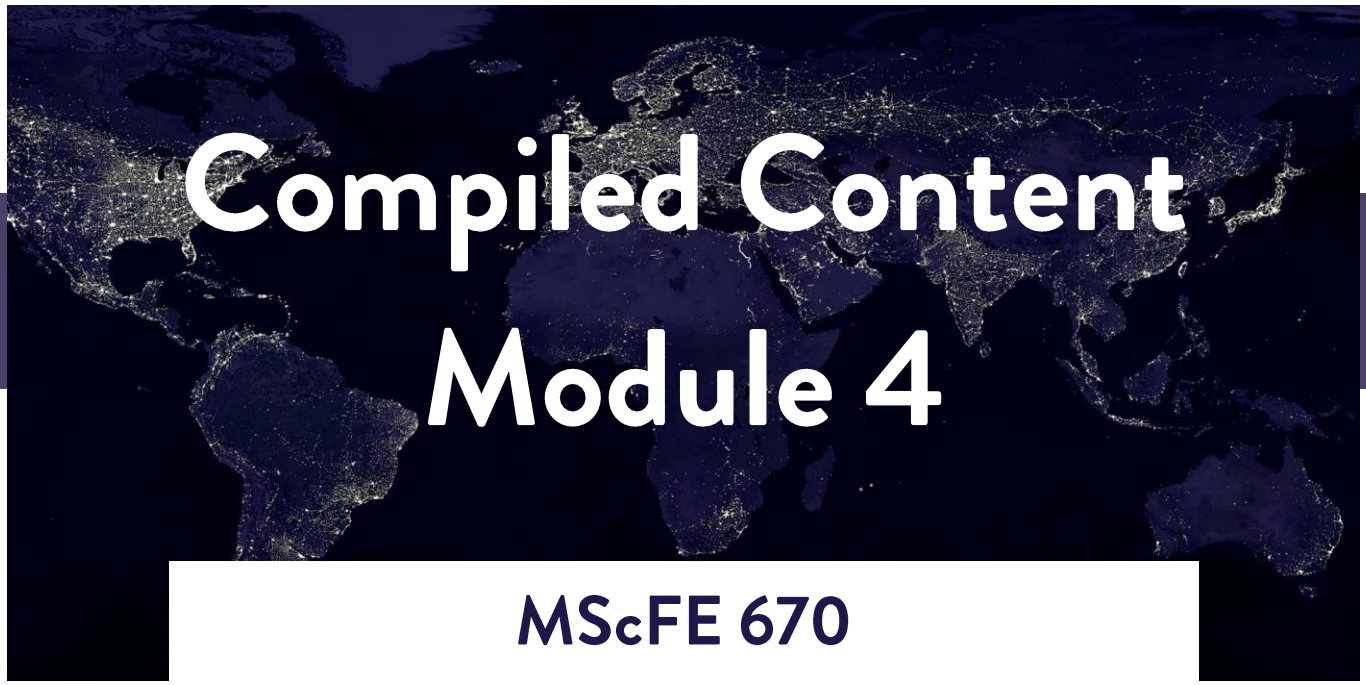
WORLDQUANT UNIVERSITY



Compiled Content Module 4

MScFE 670

Data Feeds and Technology

Revised: 09/07/2019

Table of Contents

Module 4: Blockchain (DLT)

This module begins with the definition of a blockchain and the introduction of blockchain structures to build robust, temper-proof transaction ledgers. The module continues by explaining how blockchains can be secured against double spending attacks and concludes with an explanation of the consensus protocol that is followed when more than one party holds the same transaction ledger. An example of blockchain implementation in R is discussed at the end of the module.

Unit 1: What is a Blockchain?

Introduction

Welcome to the blockchain module. Blockchain has caught the attention of the public eye in recent years, as numerous projects have started to showcase the potential of the technology. There is currently huge demand for blockchain expertise, particularly developers, across a number of industries and sectors. As the blockchain technology matures, the demand for people who understand the technology may continue to grow.

First, we will provide a quick overview of the plan for the next four modules. This module and the next one will focus on the theory of blockchains, as well as other token projects that have been created and leverage this technology. It is very important to understand these basic concepts as they will help you understand both the potentials and the limitations of blockchain. Once we have a solid understanding of the basics, we will move on to the more practical aspects. Here, we will start to create real, decentralized applications to run on blockchains, and you will learn how to code smart contracts – a potentially lucrative pursuit.

What is a blockchain?

The first blockchain built was the Bitcoin blockchain, created by Satoshi Nakamoto. The concept was first introduced in a white paper that he published anonymously in 2008. There is a lot of mystery around Satoshi, since it is still unknown who the person (or group of people) behind this name is or are.

The first concept at play in a blockchain is a *transaction ledger*. A **ledger** is a general-purpose term for some method of recording certain transactions. **Transactions** in their simplest form are events where one person sends something to another person, for example Alice sends Bob 5 USD. This is a transaction. We can record this transaction in a ledger, and make note of other transactions too – e.g. Bob sends Carol 3 USD. Hence, our ledger can help in keeping track of who has how many USD based on all the transactions detailing who sent what to whom.

Transaction ledgers have existed for a long time, in many forms, from spreadsheets to CORE banking systems. What makes blockchain different? The data in a blockchain is cryptographically secured so that any attempted change to the data it contains is easy to detect. This is why it is possible for the blockchain software to be run on many computers all around the world, without

compromising security. In contrast CORE banking systems and spreadsheets don't have the same kinds of guarantees on the data, and they are typically controlled on a single computer.

Here are four key characteristics of blockchains:

Blockchain is **complete.** This means that every transaction that has ever occurred within the system is recorded and kept forever in the blockchain.

Blockchain is **tamper resistant** – i.e. permanent**.** Once a transaction is made and recorded in the ledger it can never ever be deleted or altered. This is at least true in theory; there have been cases where human intervention was taken to change the state of the chain, be it via hacking or as an agreed resolution to a conflict. However, these cases are by far the exception, and are likely to happen less as many of these blockchain projects mature and become less experimental. As blockchains become bigger, the technology matures, and they become a more worldwide phenomenon. This property will likely be taken more seriously as more people rely on these blockchains.

Blockchain is **public**. Every transaction that ever occurs is public and can be seen by everyone. It is worth noting that you can run blockchain software privately, for example on your own computer only, but this takes away some of the advantages of using a blockchain in the first place. For example, a blockchain that helps facilitate a business process is more effective if all the parties involved in that business process have access to it – i.e. it is public to them.

Blockchain is **decentralized.** The **transaction ledger** – i.e. the list of all the transactions that have occurred in the system – is not only held by a single, centralized entity but rather many, possibly millions, have a copy of this ledger.

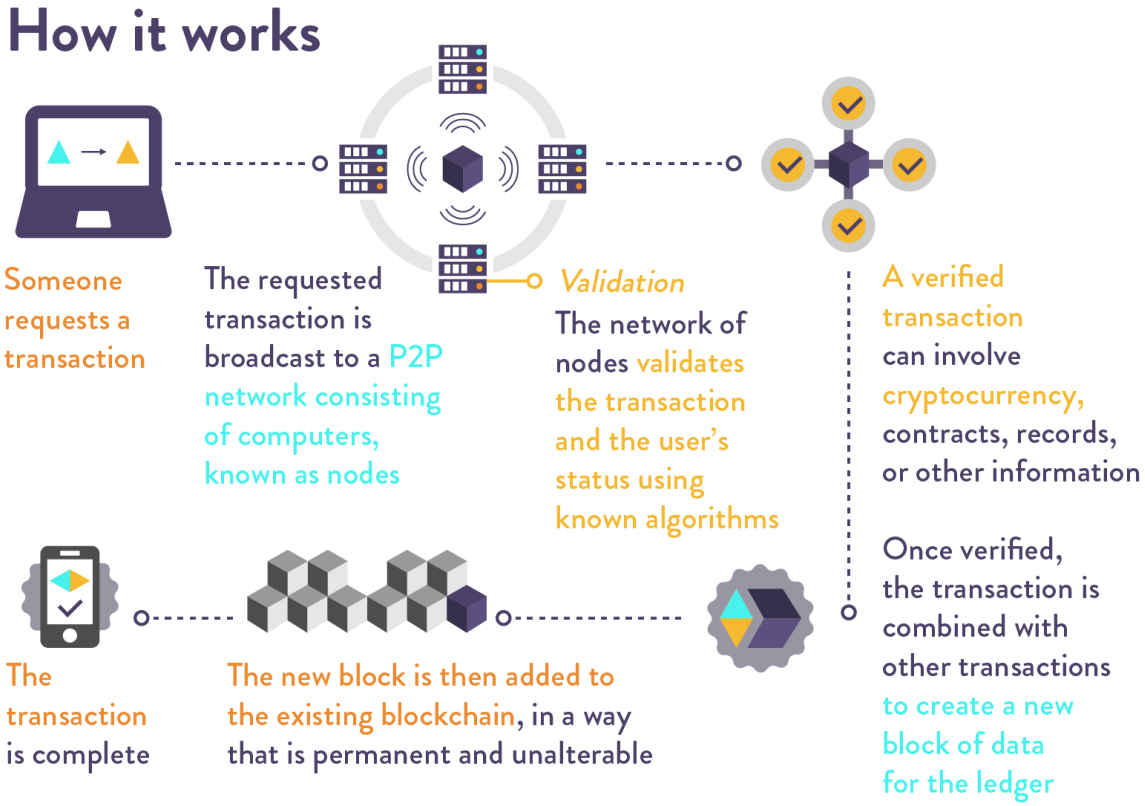


Figure 1: Blockchain basics (Source: https://www.pwc.com/us/en/industries/financial-services/fintech/Bitcoin-blockchain-cryptocurrency.html)

With these components under our belt, we can now update our definition of blockchain slightly. We can understand **blockchain** to be a complete, irrevocable, and public transaction ledger that has many copies around the world, making it decentralized. For the practical meaning of this to sink in, let's think about the implications of the above and how this may contrast to a simple ledger held by a bank.

Why use the blockchain? Why not just use a traditional ledger?

A bank's ledger will not be permanent, in the sense that, if the bank finds some transaction to perhaps be fraudulent (say, Alice sending 5 USD to Bob), then this transaction can be reversed by the bank before it is settled, since they are the single central authority that has access to this ledger. This is known as settlement risk: Bob thought he had received the funds, but the actual bank ledger entry hadn't been finalized by the bank yet. The takeaway is that the bank acts as a gatekeeper for their ledger.

Public blockchain transactions, on the other hand, have no such gatekeepers. It is completely transparent to the recipient of the payment when they received the payment and how much they received. This affords you the peace of mind that once a transaction has been made, it can never be altered or deleted. The exact details of this differ from blockchain to blockchain, and some blockchains require a period of time to pass to statistically make reverting a transaction infeasible. The flip side of this, however, is that if you make some mistake with your transaction, there is no bank you can phone to help you reverse or change this transaction.

The bank's ledger is most certainly not public. If you are Carol, you will not be able to see anyone else's transactions besides your own. Blockchains are different since they are inherently public structures. This means that you can see every transaction that has ever occurred in the entirety of the system, although not in the way you might expect. You cannot see *all* the details of the transaction. For instance, you cannot see if John sends four Bitcoin to Pam, because there is something called pseudo-anonymity that most blockchains provide. This will be covered in more detail in the next module. For now, you can explore all the transactions that have ever taken place in the Bitcoin blockchain since the first block got mined in 2008 at https://www.blockchain.com/explorer.

Finally, a bank's ledger is centralized. They are the central party that is in control of recording the transactions. Although, in reality, strict banking regulations give us the confidence that banks are operating correctly, we still have to trust them and take their word for it. This is because there is no way for us to verify the history of transactions, like we can do with public blockchains. A consequence of this is that the public may be unaware of the kinds of abuses to monetary policy taking place, and, moreover, they are powerless to prevent them. The hyperinflation that occurred in Venezuela and Zimbabwe are examples of the ways in which those in power can manipulate monetary supply.

Blockchains, on the contrary, are decentralized and transparent. This means that thousands – potentially millions – of people have a copy of all the transactions, thus making it near-impossible to delete, tamper with, or create arbitrary transactions, as it would conflict with all the copies of the ledger held by all the other people around the world. More than that, the precise number of tokens on the blockchain and the rules of how those tokens behave on the blockchain are completely defined by code.

Decentralization is hence at the core of the blockchain movement, and it is why many people are interested in it as a potential paradigm shift. It is a system where no single central authority exists with the power to change the rules of the system. Certainly, blockchains have the potential to cause a paradigm shift, but what that paradigm shift will involve and how it will take place is unclear. This is part of the reason why different groups of people – from libertarians, to anarchists,

to socialists – have tried to grasp how this technology could fit into shaping the future of human society.

Summary

This does not mean that blockchain is a silver bullet for all problems, though; it is still very much in its infancy. There are still numerous issues facing blockchains today, and a multitude of technological advances are still needed before true mainstream adoption could occur. Even then, there will likely always be a good reason for the centralization of certain things. Blockchain technology has caught the imagination of many, and opinions on the technology and what it means for humanity vary greatly. The above arguments are presented, not to choose sides or point fingers at governments, but rather to expose you to some of the common thought around the topic, and to explain why so many people are diehard blockchain fans. The following modules will strengthen your understanding of blockchain and prepare you to consider and explore blockchain best practices for the future.

Unit 2: Blockchain Structure

Introduction

Now that we have a high-level understanding of what blockchains are, let's dive a bit more into how they actually work. Remember: a blockchain is a *complete, permanent, public,* and *decentralized* transaction ledger, which can be broken down into several components:

1. Hashing (cryptography)
2. Private/public key cryptography
3. Merkle trees
4. Consensus (BFT, PoW).

Hashing

An important concept in blockchain is that of a hash. A **hash function** is simply a function that takes data as input and outputs different data in a deterministic manner – i.e. with the same input it will always produce the same output. The hash functions used in blockchains have three particular security properties. Here is the more mathematical definition for a hash function H, followed by a more colloquial explanation:

1. Collision-free, infeasible to find distinct x and y such that $H(x) = H(y)$.
2. Hiding, given $H(x)$, it is infeasible to find the value of x.
3. Puzzle-friendly, for all output y, k chosen from a high min-entropy distribution, it is infeasible to find x such that $H(k|x) = y$.

Collision-free means that it is hard to find two values that hash to the same value. **Hiding** means that, given a hash, it is difficult to find the value that gave you that hash. Finally **puzzle-friendly** means that it is hard to find a piece of data to append to a given piece of data to give a desired hash. Puzzle-friendliness is important for proof-of-work mining, which is explained a bit more later in the course, so please don't linger too long on this third property – it is mentioned for completeness of the definition.

Ledger

A blockchain is made up of blocks, which are hashed together, and each block is made up of hashes of transactions (see Figure 2):

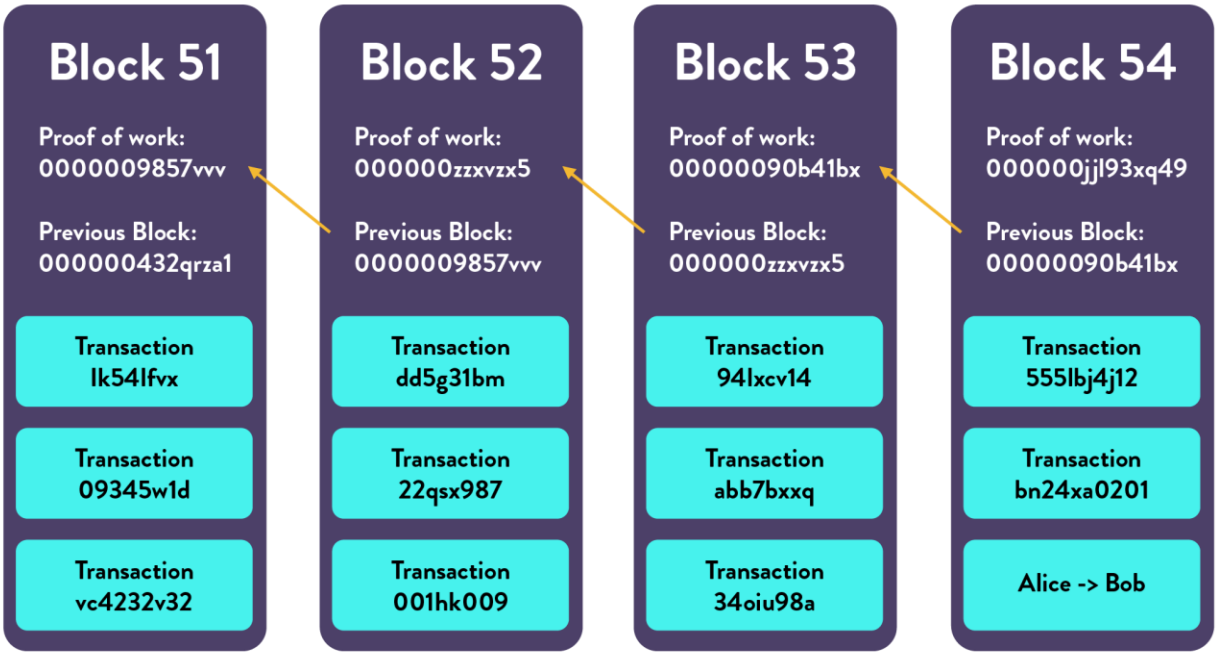


Figure 2: Blocks in a blockchain (Source: http://www.ybrikman.com/assets/img/blog/Bitcoin/Bitcoin-block-chain-verified.png)

Blocks contain a group of transactions, and delineate time. For example, transactions in block 52 occur before block 53 and after block 51. All transactions within the same block take place (or took place) at the same time. Transactions are therefore grouped into blocks and added to a chain: hence a "blockchain". Chaining these blocks and transactions together like this ensures they are ordered and remain cryptographically secure.

Getting free pizza

Imagine we want to buy some pizza with Bitcoin, but we want to cheat the system and get the pizza for free. What possible plans of attack might we employ and how does the blockchain ensure we cannot get our free pizza?

Our first option is to *try to spend money that is not ours.* We might try to create a transaction to pay the pizza shop but use the public address of someone else. This is not possible, however, and the reason why has to do with public and private keys.

Public and private keys

Consider this: you are given two unique numbers. The first number, 0xa912bd66cEA7 f8c2c20FC45Df7f042Ce084344F7, is publicly known to everyone. This is called your **public key** and it functions much like a bank account number. The second number, bdfd18d241d01fd4b25f5d1ffe54dfd5991f63e8d4e8ceec2ece370281f03ee2, is known solely by you. This is called your **private key** and we can think of it as the password to your bank account. Never ever reveal your private key to anyone.[1] These two numbers are mathematically related in practice in such a way that only you know the private key and will be able to create valid transactions from the corresponding public address.

Anyone can easily generate a public and private key pair for free (more details on this in the next set of notes).

You can give your public address to anyone and they can easily send funds to you. With funds in your public address, you may want to create a transaction sending money from that public address to another public address, say, our pizza shop in question. To do this we have to create a digital signature we can use to sign the transaction in question. Using our private key that relates to our public address, we create a digital signature and sign the transaction. Now anyone can easily – i.e. in computationally inexpensive manner – verify that the digital signature created was created from the private key relating to the public address in question. The mathematics of how this actually works is beyond the scope of this course, but you can read up on how this works mathematically by visiting the source provided in the Further Reading section.

In summary, thanks to cryptography, we can now be sure that only the person who holds the private key related to a particular public address can generate valid transactions from said public address. Therefore, we cannot get a free pizza by trying to create transactions sending money from someone else's public address to the pizza shop since *we do not know their private key.*

[1] This account has no funds, so there is no risk in sharing it as an illustration.

Merkle trees

Why do we not just go back in time to change a transaction from long ago, and give ourselves a little more money, then?

Recall the structure of a blockchain as depicted in Figure 2. In each new block created, we must hash it and include it in the next block. More specifically, we hash the header of the previous block along with Merkle root in our block to get the header of our current block. A cryptographic, one-time value called a **nonce** is also included in the header of Bitcoin; the reason for this will be elaborated on in the last section of this module. But what does this mean in layman's terms?

Remember that, currently, each block contains several transactions, all of which are valid since we use their digital signatures to check their validity. These transactions are stored in something called a Merkle tree, which you can see in Figure 3.

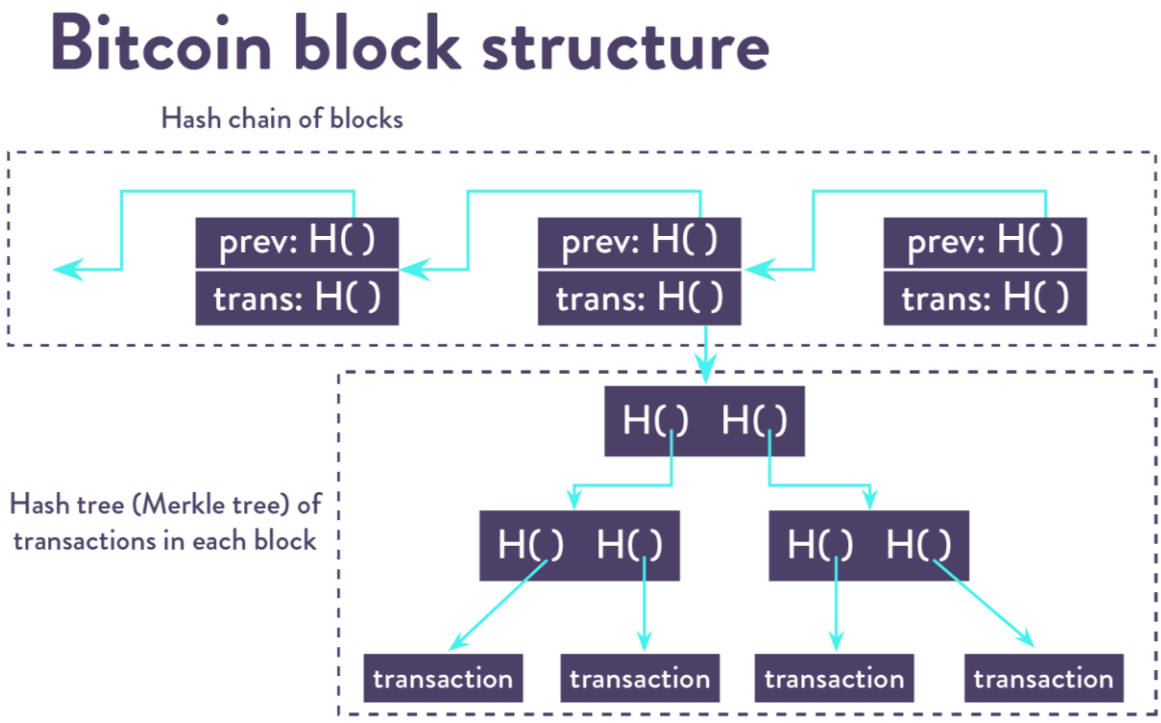


Figure 3: Merkle tree

A **Merkle tree** is simply a hashed binary tree. **Binary trees** are data structures in which a record is linked to two successor records, and they are hashed when each of these records is a hash. Binary trees are very useful data structures that allow us to have faster search times, compared to us

keeping everything in a simple list. **Hash functions** are functions that map arbitrary data to data of a fixed size. An example of a hash function would be:

H("Learning about hash functions is interesting") -> d38fh7gaff9af0a

In practice, our transactions in the Merkle tree are hashed, and then the hashes at each superior level in the tree are hashed, right up until we have a *Merkle root*. An important property of hash functions is that they have something called an **avalanche effect**. This means that even if we were to just change one little thing in the above example, like adding a full stop:

H("Learning about hash functions is interesting.") -> h6uw8q0jk673fg27 ,

our hash function would create a completely different hashed value, making it seem completely uncorrelated with the previous hash value. Therefore, if we tried to go back and alter a transaction in a previous block to get some more money for pizza, any change at all to a previous transaction would completely change the *hash* of that transaction. This, in turn, would change the hash of the *Merkle tree* right up to the *Merkle root*. Which would then change the hash of the entire block and therefore the hash of all subsequent blocks since for each block we hash the header of the previous block. At a very high level of analysis, this shows how there is, therefore, no possible way to go back and manipulate transactions in the blockchain to benefit yourself, due to the properties of the hash function described at the beginning of this section.

Hashing, *Merkle trees*, *Merkle roots*, and the *avalanche effect* all help answer why the blockchain is structured as it is. A blockchain is a chain where each block contains the hash of the previous block, and this ensures that the system is completely immutable: no-one is able to tamper with previous transactions.

Summary

So far, no free pizza. We have explored the techniques implemented by blockchains to ensure a robust, tamper-proof transaction ledger. In the next sections, we will continue to answer more important questions concerning blockchains like: Who gets to the add blocks?

Further reading

https://Bitcoin.stackexchange.com/questions/67486/private-keys-transaction-verification
https://medium.com/s/story/how-does-the-blockchain-work-98c8cd01d2ae
https://bitsonblocks.net/2017/09/04/a-gentle-introduction-to-interbank-payment-systems/

Unit 3: Double Spending Attack

Introduction

In the previous set of notes, we started to think of ways we could get a free pizza by cheating the blockchain system. Both attempts we tried so far were unsuccessful, but they illustrated some of the important mechanisms present in blockchains that ensure their constant integrity. We will now turn our attention towards another type of attack that blockchains need to be secure from: double spending.

Double spending

Imagine that a pizza cost 1 Bitcoin and that we had exactly 1 Bitcoin to spend. We could create one transaction where we sent 1 Bitcoin to pizza shop A and at the same time create another transaction where we also sent 1 Bitcoin to another pizza shop, pizza shop B. What we are trying to do here is get two pizzas by **double spending** our 1 Bitcoin. If each transaction were done on its own then it would be completely valid since we do have 1 Bitcoin to spend. How then, does the blockchain prevent us from double spending our 1 Bitcoin? We first need to introduce a little a more theory before we can sufficiently answer this question.

Remember the structure of a blockchain as illustrated below:

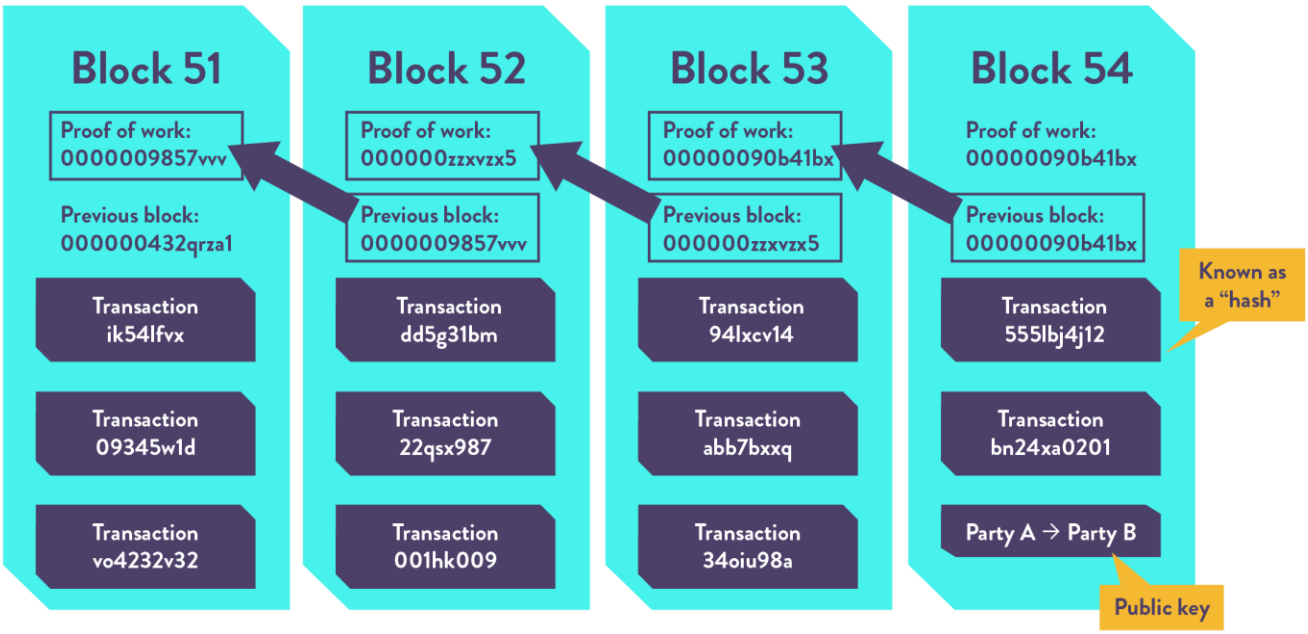


Figure 4: Blocks in a blockchain (Source: JP Morgan report)

We group transactions into blocks and add these blocks to the end of the chain, with the order of the blocks delineating time. Recall that a blockchain is a decentralized system, and hence no central authority has control of the transaction ledger. A consequence of this is that different people will be adding blocks to the end of the chain all the time. Those that do this are called **miners**, and the process of adding blocks to the blockchain is termed **mining**. This is why you may often hear people talking about "mining cryptocurrency". In fact, all they are doing is attempting to add blocks to the end of the blockchain. Later we will discuss some of the economic incentives of doing so and how this mechanism functions in practice, but for the meantime let's revert to talking about the mining process at a high level.

Whenever a transaction is created, it is broadcasted across the network and put into something called the **unconfirmed transaction pool**. These are all the new, proposed transactions that are yet to be **confirmed** – i.e. put into a block and actually added to the blockchain. Miners select transactions from this unconfirmed transaction pool and propose a new block to be added to the end of the chain. We will cover the mechanics of this process in more detail later, but for now, let's shift our attention back to our double spending attempt to get two pizzas.

In an attempt at double spending, we have created two transactions. Each transaction sends the same 1 Bitcoin to a different pizza shop in an attempt to double spend our 1 Bitcoin. For the sake of simplicity, we will refer to the transactions as transaction X and transaction Y respectively. Now imagine a miner, Sam, who is including transactions in her next proposed block. She may include

either transaction X or transaction Y, but she would not include both transaction X and transaction Y, since including both would amount to a double spend, therefore rendering her new proposed block invalid. Since it is an invalid block, no other miner would want to build on top of it, and thus she would get no reward from producing that block, as the community would ignore it due to its invalidity. Knowing this, Sam only includes transaction X (or only transaction Y) in her proposed block.

However, we could have another miner, Jane, who includes transaction Y in her proposed block. Transaction X and Y are now both included in proposed blocks. It is important to note that miners compete to get their proposed blocks added to the chain. The most common method for this competition by solving a computationally intensive puzzle in something called proof-of-work, which we will discuss more later. For now, let's just imagine that the miners, Sam and Jane, both solve the puzzle at the same time, and they both add valid blocks to the chain.

Sam, then, has added block B1 to the chain, and Jane has added block B2 to the chain, as depicted below:

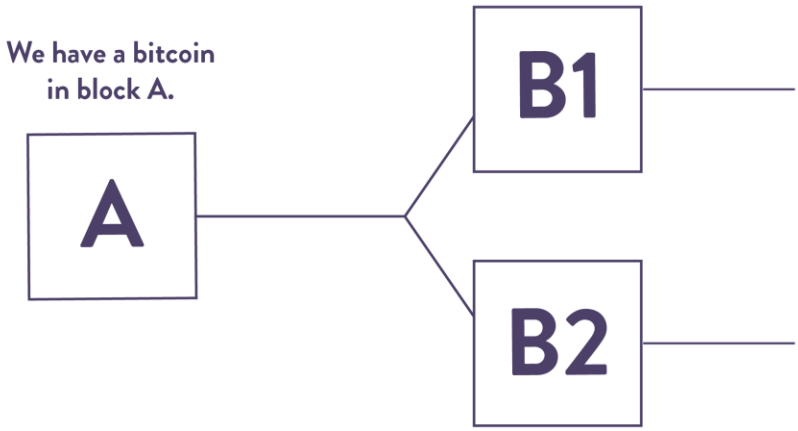


Figure 5: Sam and Jane's blocks

Both are valid blocks and thus create a temporary fork in the blockchain. There is a rule stating that miners always add blocks to the longest current chain. Miners would now continue the process of trying to add a block to either the chain A-B1 or the chain A-B2.

So far, we have succeeded with a double spend. We have transaction X included in the one fork of the chain and transaction Y included in the other fork of the chain. Nevertheless, this success is short-lived.

When a block is added to a chain, we follow our rule where miners continue to add blocks to the longest current chain. Therefore, when block C is added to A-B2, the longest chain is now A-B2-C, and the earlier block B1 is said to be **orphaned**. Figure 6 depicts this.

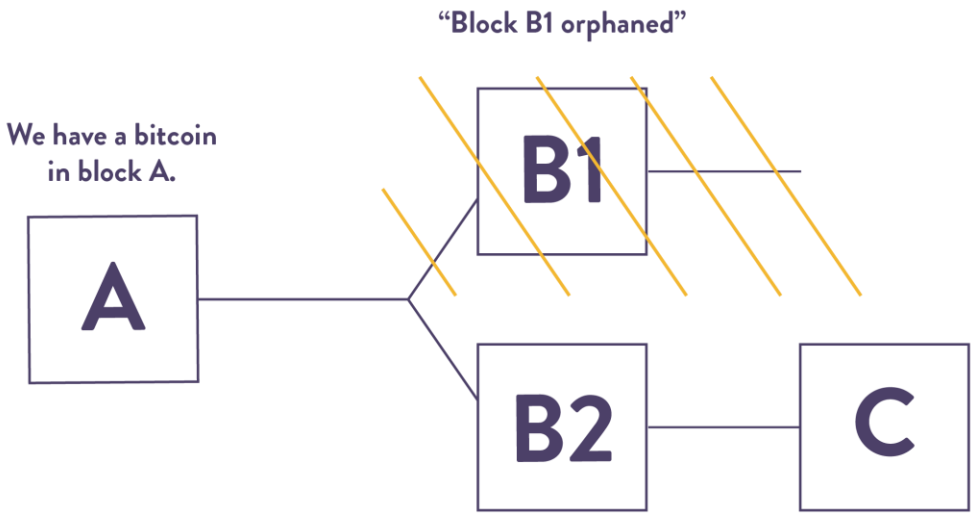


Figure 6: Orphaning

Orphaning in our example here means that all the transactions contained within block B1 are no longer included in the chain and instead return to the unconfirmed transaction pool where they may be included again in a later block.

What does this mean for out attempted, pizza double spend? This means that our one transaction would have been successfully included in block B2. Our other transaction in block B1 would now have returned to the unconfirmed transaction pool.

Since we have already spent our 1 Bitcoin on pizza in block B2, our other transaction trying to spend that same Bitcoin on another pizza will be invalid due to insufficient funds. It will therefore not be included by any miner in newly proposed blocks. At a very high-level of analysis, this is how blockchains successfully defend against double spend attacks.

What does this mean practically for the pizza vendors, though? Well, in the initial case, both transactions trying to spend the same Bitcoin were included in the blockchain. Each pizza vendor saw a valid transaction with 1 Bitcoin coming into their address, and if they both immediately sent pizzas, one vendor would lose their payment at a later stage when the one block containing their transaction gets orphaned.

While, in reality, it depends on which blockchain we are talking about, it is unlikely that two blocks would get added to the same chain at the same time, thus creating a temporary fork. However, it does happen. As a general rule of thumb, in the Bitcoin blockchain merchants wait for approximately three confirmations. This means they wait for three additional blocks to be added onto the chain that builds upon the block that contains their transaction. This makes it highly unlikely that the transaction will be orphaned, and subsequently render the merchant without payment. In Bitcoin, blocks are added to the chain every 10 minutes on average. This equates to merchants generally waiting 30 minutes (on average) before assuming the transaction to be final.

This presents a current limitation of the blockchain. Imagine going to buy a drink with some Bitcoin and the vendor makes you wait 30 minutes to check your payment is final.

This is just one reason why Bitcoin is likely not ready to replace traditional currencies quite yet. Many advances still need to take place before a blockchain can function in a mainstream, global context as a payments system on par with traditional currencies, although some will disagree with this. It should be noted that there are indeed other blockchains that do things differently, and which have improved upon the current limitations of Bitcoin. Many blockchains have different trade-offs – e.g. decentralization vs. scalability, which we will cover in the context of EOS and Ethereum in the next module – and it is up to the person using the blockchain to decide what is best for the particular needs in question.

Summary

To summarize, we have investigated multiple ways to try and get a free pizza by attacking the integrity of blockchains. From this, we have highlighted several key concepts such as digital signatures, Merkle trees, hashing, and mining, all of which, combined together, ensure the integrity of the blockchain system.

Unit 4: Consensus

Introduction

A consensus is a general agreement among a group of people. Recall that a blockchain is a decentralized ledger with many different parties holding copies of this ledger around the globe. **Consensus** in the context of blockchain is therefore a general agreement as to the state or contents of the transaction ledger among the group of people who hold copies of the ledger.

Let's assume that, currently, all parties hold the exact same transaction ledger. Naturally, as new transactions occur, people all around the world need to update their transaction ledger and reach "consensus" as to the state of the transaction ledger. Given the scale of this task, this is not that easy. A solution needs to be created as to the best way to achieve this consensus in a secure manner. Different blockchains implement different consensus protocols, many of which have different trade-offs. In the next section, we will outline the idea behind the consensus protocol "proof-of-work", currently used by Bitcoin. When we look at other blockchain systems like Ethereum and EOS in the next module, we will investigate other types of consensus protocols.

Consensus has been a topic for a long time in computer science. For example, Facebook will have a consensus algorithm to ensure that the photo that you uploaded eventually ends up in all their data centers, so that people from all around the world see the same image. What made Bitcoin and its proof-of-work algorithm such a breakthrough is that it is completely public and permissionless – i.e. anyone can join the network. All consensus algorithms that existed before Bitcoin were permissioned, and entities in these networks had given roles. In Bitcoin, miners can join and leave the network at will and the network continues to operate correctly.

Public blockchains are decentralized, transaction ledgers. They are novel because the concept of decentralization ensures a "trust-less" system, where no single entity has the power to control or manipulate the system. In order to securely implement a decentralized transaction ledger, where anyone in the world can have a copy of this ledger and validate it, we need a **consensus protocol** that will consistently ensure agreement among all parties as to the state of the ledger.

Proof-of work

Let's now imagine that we are in a state of consensus. As we have seen before, as new transactions occur in the system, they are all added to the unconfirmed transaction pool, waiting to be included into a new block and appended to the chain. The problem of consensus boils down to how we add

blocks (groups of transactions) to the chain while getting all parties involved to agree on this new state of the blockchain when this new block is added, and on which transactions are included in the block.

Since the idea behind blockchain is for it to be a decentralized system, we want to ensure that no single entity has the ability to dictate which transactions are added to blocks in the chain, as this would amount to centralization. Rather, the idea of Satoshi Nakamoto, the creator of Bitcoin, was that everyone in the world should have the chance to add blocks, ensuring maximum decentralization. He termed this "One CPU, one vote" in the Bitcoin whitepaper.

His idea – i.e. proof-of-work – was that computers around the world (miners) compete with their computational resources (CPU) to solve a computationally intensive puzzles, with the first person to solve the puzzle getting to add their proposed block to the chain. Whenever someone solves this puzzle, they have found what we call a **valid block**. They can then add this block to the chain and broadcast this block to the rest of the network. The rest of the network can then easily confirm that the block is indeed valid and add it to their version of the blockchain, and thus consensus on this new block is eventually reached among the participants.

To be more specific, we need to dig up the third part of our definition for a hash function: that it is puzzle-friendly. Miners take the data of the block and search – i.e. guess randomly – for a nonce such that the resulting block header has the property that it is smaller than a certain threshold that is agreed on by the network. This threshold is tuned such that Bitcoin blocks occur on average every 10 minutes, and Ethereum blocks occur roughly every 14 seconds. You can see the block headers in any block-explorer and notice that they always start with a few '0's, as seen highlighted in yellow in the following diagram.

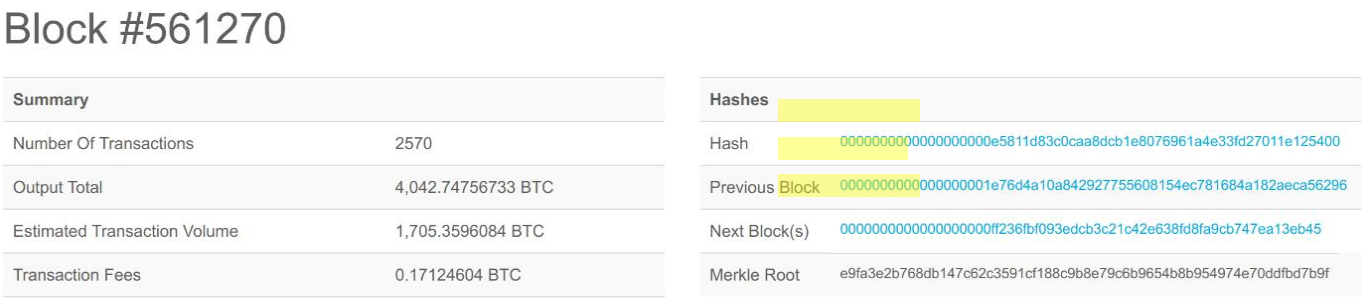


Figure 7: Block headers

How are miners incentivized to use their computer to add blocks to the chain? If a miner finds a valid block, they get something called a **block reward**. This is essentially a transaction sending them some Bitcoin, or, more abstractly, a financial incentive to mine.

From a game-theoretic point of view, this block reward incentive is extremely important in the whole process. There is nothing preventing a miner from rejecting a block mined by another computer, and trying to mine their own, different block. However, if that block that they mine never gets included in the blockchain – i.e. if the rest of the network don't come to consensus that that block is to be included in the blockchain state – then that miner will never get the block reward, since it is not part of 'The Blockchain" that the rest of the participants agree on. So, miners are incentivized to always mine on the longest chain (of blocks).

This mining incentive is also important as a fork prevention mechanism.

A fork occurs when different portions of the network agree on a different set of blocks to be added to the chain, as seen in the below diagram. However, the two parts of the chain will not grow at the same speed forever. Miners want to be on the winning side of the fork (so they don't want to lose their block reward), so once again miners will always join the longest chain, effectively killing the fork.

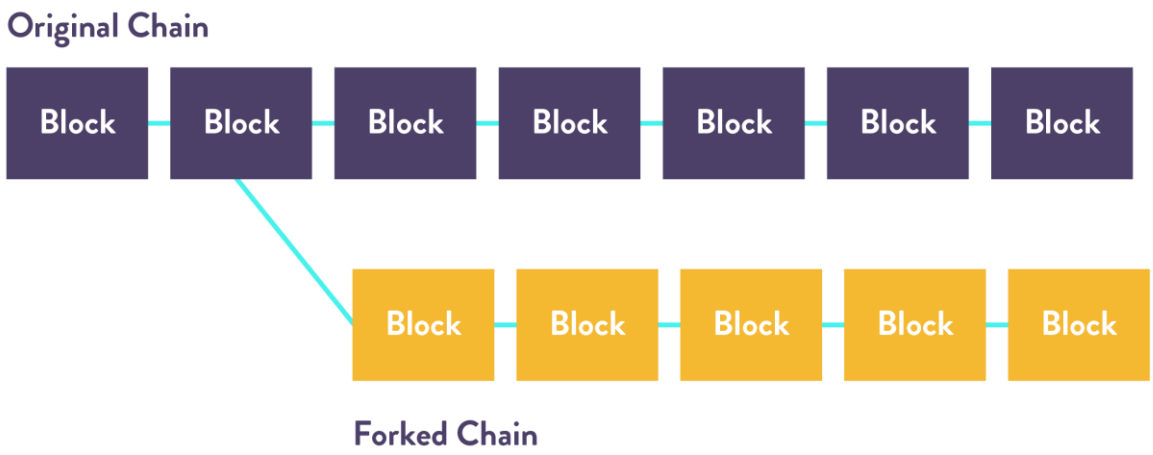
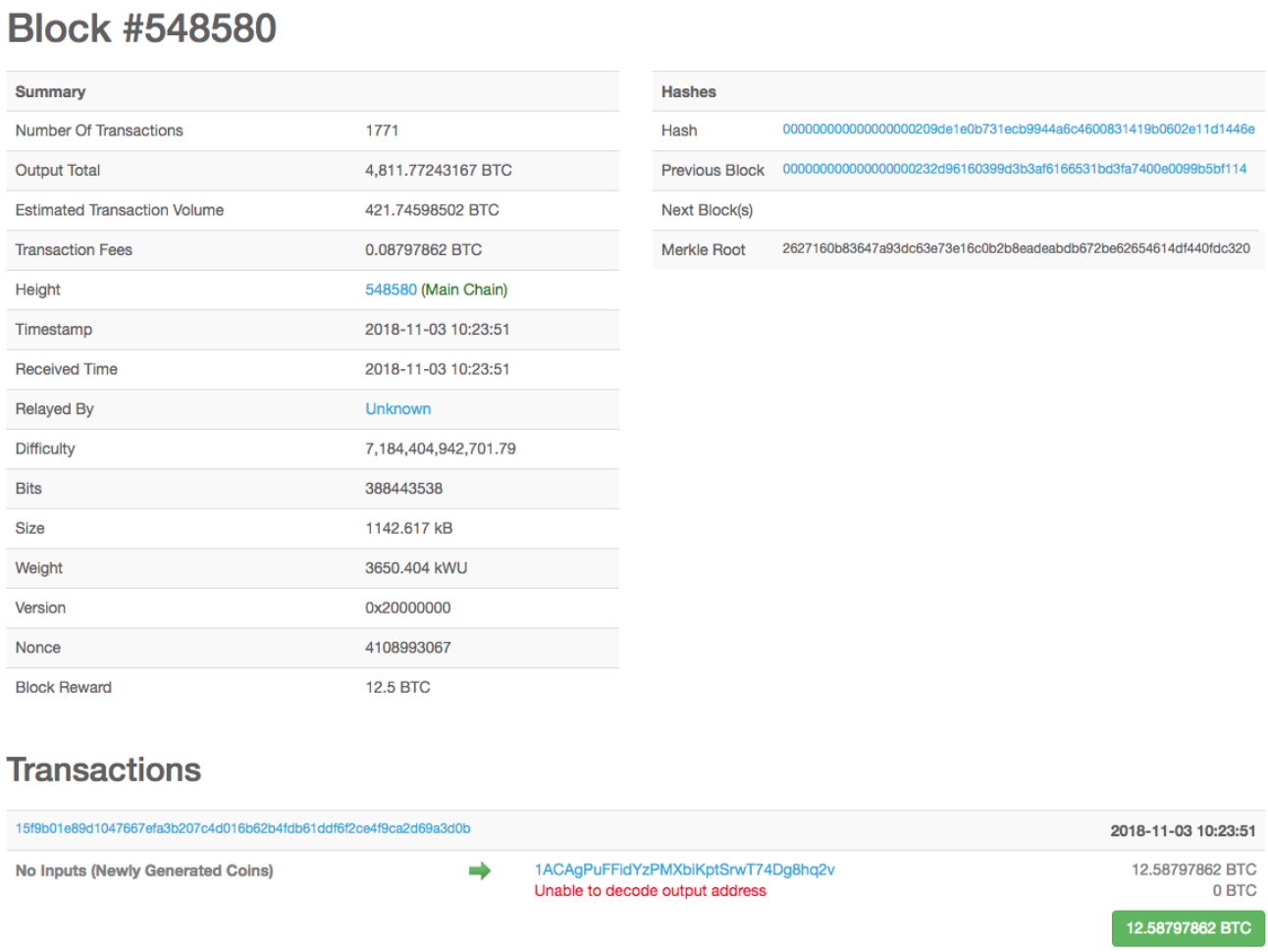


Figure 8: Fork

As an aside, a related concept is that of a hard fork, which is when the network is permanently partitioned after the fork. This is normally done for software upgrades, when the old software would no longer be compatible with the new versions. Sometimes both chains continue to exist when ideological conflicts arise about the technology; other times the old chain stops existing as the vast majority of the miners support the software upgrade.

Have a look at the following Bitcoin block (Figure 9):



Block #548580

Summary	
Number Of Transactions	1771
Output Total	4,811.77243167 BTC
Estimated Transaction Volume	421.74598502 BTC
Transaction Fees	0.08797862 BTC
Height	548580 (Main Chain)
Timestamp	2018-11-03 10:23:51
Received Time	2018-11-03 10:23:51
Relayed By	Unknown
Difficulty	7,184,404,942,701.79
Bits	388443538
Size	1142.617 kB
Weight	3650.404 kWU
Version	0x20000000
Nonce	4108993067
Block Reward	12.5 BTC

Hashes	
Hash	00000000000000000209de1e0b731ecb9944a6c4600831419b0602e11d1446e
Previous Block	00000000000000000232d96160399d3b3af6166531bd3fa7400e0099b5bf114
Next Block(s)	
Merkle Root	2627160b83647a93dc63e73e16c0b2b8eadeabdb672be62654614df440fdc320

Transactions

15f9b01e89d1047667efa3b207c4d016b62b4fdb61ddf6f2ce4f9ca2d69a3d0b			2018-11-03 10:23:51
No Inputs (Newly Generated Coins)	→	1ACAgPuFFidYzPMXbiKptSrwT74Dg8hq2v Unable to decode output address	12.58797862 BTC 0 BTC
			12.58797862 BTC

Figure 9: Random Bitcoin

If you bring your attention to the last field, you can see that it shows the block reward of 12.5 BTC. This means that a miner earns 12.5 BTC currently for every block he adds.

If you look at the transaction in Figure 9, you will notice that it says, "No Inputs (Newly Generated Coins)". So where do these block reward coins come from? The total supply of Bitcoin will be 21 000 000 BTC. At the time of writing, the current circulating supply is around 17 400 000 BTC. As miners mine blocks, they receive Bitcoin as a *block reward,* which effectively increases the circulating supply toward the total supply. Every 210 000 blocks, the block reward halves. Given that Bitcoin blocks are found, on average, once every 10 minutes, this equates to roughly every four years. Therefore, in 2020 we should see the block reward halve to about 6.25 BTC per block. Once the total supply is reached, no more block rewards will be given to miners. The logical next

question is: What happens then? Why would the miners continue mining if there is no financial incentive to do so?

The short answer boils down to transaction fees. However, no one knows for sure if transaction fees alone will be enough to secure the network: only time will tell. Ethereum, on the other hand, has taken the approach of always having a block reward to ensure the network stays healthy far into the future.

When anyone broadcasts a transaction, they attach a chosen *fee* at their discretion. Miners then select transactions from the unconfirmed transaction pool to include in their proposed block. Miners are financially incentivized to include transactions with the greatest fees attached to them in their proposed block as the fee also gets given to the miner. This mechanism is already in practice, and it is why people with urgent transactions attach big fees in order to incentivize miners to include their transaction in the next possible block.

Proof-of-work shortcomings

Where does proof-of-work fall short, though? Due to the financial incentives of mining Bitcoin, people over time have used purpose-built GPU, and now ASIC chips, to solve the computationally intensive puzzle much quicker than before, giving them a greater chance of adding a valid block and claiming the block reward. Because specialized mining hardware (such as ASIC chips) are required to even have a small chance of finding the next valid block nowadays, the principle of "one CPU, one vote" is no longer applicable. This presents a significant barrier to entry and reduces the decentralization of the ecosystem, as now only certain individuals with expensive mining hardware stand a chance at adding valid blocks. In addition, there are also major environmental concerns around the significant amount of electricity used in the mining process to simply solve a computationally intensive puzzle – something that does not directly add any value to society.

Decentralization is reduced even more as people join **mining pools**, which function much like lottery clubs do. Since an individual's chances of adding the next block to the chain are miniscule, and yet there is such a big reward, adding a block has become similar to a lottery where your chances of winning are very small. People thus join mining pools where they agree that if any one of them find the next block, they will split the reward with the entire mining pool. This ensures a more stable source of income for miners. The drawback, however, is that large mining pools that control much of the total mining power have too much influence over the blockchain, rendering it as more of a centralized, and less of a "trustless", system.

One of the drawbacks to the decentralized and trustless nature of public blockchains is that they are hard to scale. How do you find ways to increase the throughput of the system without introducing a higher degree of centralization? Bitcoin can currently handle about 7 transactions per second, and Ethereum around 15. Visa, on the other hand, say, "Based on rigorous testing, we estimate that VisaNet is capable of processing more than 12,000 transaction messages per second"[2]. This is known as the scaling problem of blockchains, which we will look at briefly later in the module when we start examining other consensus protocols.

Summary

In this section we learned that, since blockchains are decentralized ledgers, a mechanism to ensure agreement on the state of the ledger by all parties is necessary. Bitcoin's proof-of-work consensus protocol ensures the consistent state of the transaction ledger through all parties aiming to solve a computationally intensive puzzle in order to add a valid block containing new transactions onto the blockchain. Proof-of-work has its drawbacks, and many other consensus protocols are employed by other blockchains, each with their unique trade-offs. We also saw a small implementation of Blockchain using R code to illustrate the concepts that underlie the technology.

In Module 5, we will examine other consensus protocols.

[2] "Visa Inc. at a Glance." https://usa.visa.com/dam/VCOM/download/corporate/media/visa-fact-sheet-Jun2015.pdf. Accessed 18 Feb. 2019.

Bibliography

"Making sense of Bitcoin and blockchain: PwC." Accessed February 1, 2019. https://www.pwc.com/us/en/industries/financial-services/fintech/Bitcoin-blockchain-cryptocurrency.html.

Narayanan. A., Bonneau. J., Felte. E. W., Miller. A., Goldfeder. S., and Clark. S. 2016. *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, New Jersey.

Collaborative Review Task

In this module, you are required to complete a collaborative review task, which is designed to test your ability to apply and analyze the knowledge you have learned during the week.

Overview

Your task is to use your understanding of blockchain to produce a 1-2 page-long research report on EOSIO software.

Brief

Your report should be able to give any investor a quick overview of the project, covering the potential risks, benefits, and any relevant circumstances concerning the project's success. The report should be clear, focused, and provide a compelling argument as to the claims made therein. Based on your preliminary report, an investor should be able to quickly decide whether to further investigate this project or not.

Your report should explain the following aspects of the project:

1. A basic overview of the project, including its development and key features.
2. The economic model of the project, such as the types of markets it could disrupt, potential revenue streams, and whether it makes sense for adoption by business (now or in the future).
3. Any significant challenges you think the project may face, such as adoption or scalability.

Your report should include rationale and sources for these points. One resource that may be of use is https://coinmarketcap.com/. From the homepage, you can scroll down and click on a project. From there you can click on the link to their website and start to learn more about this project, being sure to take note of the project's whitepaper.