LISTEDRESERVE

Part 1: What Is Bitcoin?

September 2019

Research Note from ListedReserve

This short series sets out to explain Bitcoin. In some cases, we will go into detail to explain some of the genius incorporated into the protocol that is little understood. It remains to be seen if Bitcoin will be a success, but in reading this series, you will realise that some of the great discoveries in modern mathematics and computer science are at work here.

It should take half an hour to read. Go slow and understand the maths, which is basic and worth the effort.

Bitcoin, as you will see, exists at the intersection of computer science, advanced mathematics, game theory and Austrian economics.

Contents

1. How This Document Works

Our aim here is to deliver a relatively full explanation of how Bitcoin works

Summary Box

The orange boxes will contain a couple of lines explaining the point without the mathematical details.

Technical Box

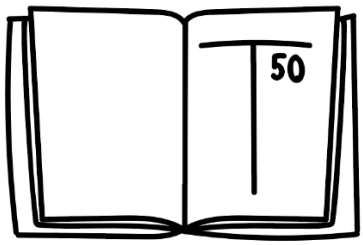
The blue boxes will highlight the important parts of the technology. You might consider this the 'invention' part of Bitcoin.

2. What Is Bitcoin?

Bitcoin is a tamper-proof ledger recording ownership, nothing else.

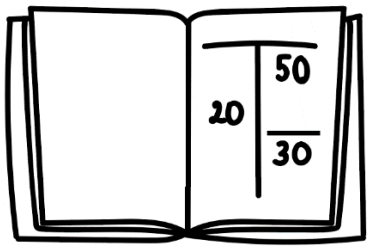
3. A Record of Ownership

Starting with banks, let's discuss the traditional system with which we are all familiar. A traditional bank is simply a ledger keeper. If you deposit $50, then the bank records that it owes you $50.



You are relying on the bank to get the record correct, to back it up, and to keep it secure.

If you want to send the money, you must contact the bank. It will check the ledger and confirm your balance, and providing it approves of where you are sending it (so not Iran, North Korea, etc.), it will send your money to another bank. That bank does the exact same thing for the receiving customer.



The whole process is just movements on a ledger—$50 in, $20 out, balance of $30—all facilitated by a middleman who is paid for the trouble.

4. Bitcoin's Ledger

Bitcoin is exactly the same as a traditional bank. It is a ledger of transactions. Everything is simply a record of inputs and outputs and who owns what. When you hear 'blockchain', hear 'ledger'—a record of ownership.

Bitcoin is as simple as that.

Bitcoin
Bitcoin is just a list of transactions and their owners. That is all the blockchain contains.

5. The Problem with Internet Money

Internet money faces the same issue as other software. It can be counterfeited easily—for example, music and Microsoft Windows can be copied infinitely.

Internet Money and the Double-Spending Problem
Without a bank, if you receive money on the internet, how do you know the person you're sending it to you a.) Actually has it? b.) Hasn't already spent it?

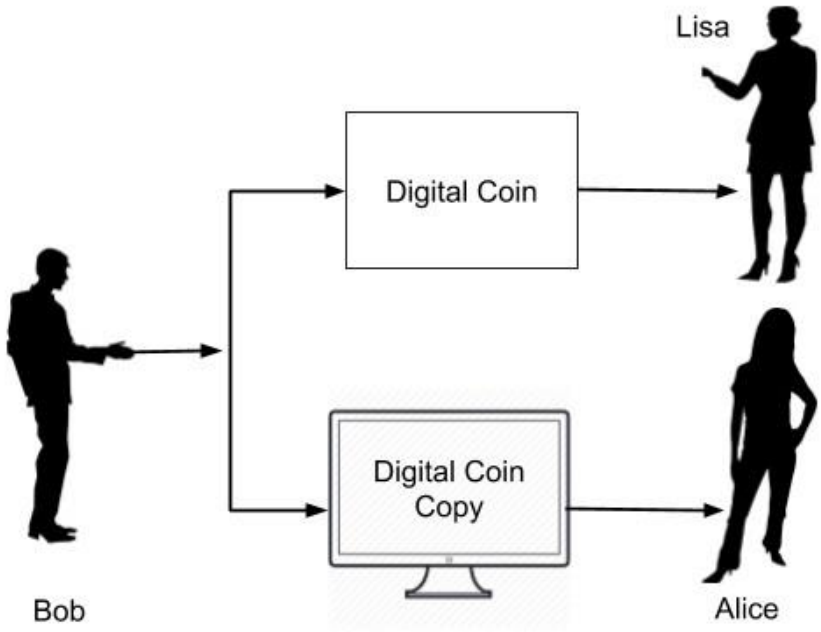
With internet money, there is no trusted middleman. In banking, we know the bank has checked that the person sending us money has the money and has not sent it elsewhere. We do not know that anyone has checked that the money exists on the internet.

The double-spending problem is as follows:

- A Has $20.
- A pays B $20.
- A also pays C $20.
- C and B do not know that A has paid them both $20.

The double-spending problem was one of the longest-standing problems in computer science.

On the internet, you would not know if a person sending you digital money has sent that same money (counterfeit money) to another person.

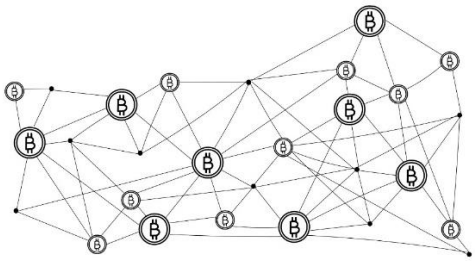


6. Solving the Double-Spending Problem

Solving the Double-Spending Problem
Bitcoin uses a network of users who watch everyone else's activity. If someone attempts to spend money a) That they do not have Or b) That they have already spent their transactions are rejected by the network.

Rather than have a middleman, Bitcoin uses every user to check that transactions are valid and that digital money has not been spent twice. Here, we will break down every step of this process in this series, step A–E, although they do not necessarily happen in that order:

A. Bitcoin is a network of computers, like the internet. All network members keep a copy of the ledger and validate transactions on it according to the rules in the software.



B. Transactions take place on Bitcoin's blockchain.
C. Bitcoin uses miners to check transactions every ten minutes to make sure nobody has spent coins twice.
D. Mining is a competition. The miner who wins gets to report the correct transactions in a new block and award themselves some new bitcoins.
E. Everyone on the network can easily check if the miner's answer to the puzzle is correct.
F. The reward for the winner is that they get new bitcoins.

7. The Tools of Bitcoin

First, we need to explain some special features of Bitcoin that will make further explanations easier.

Many people believe Bitcoin is very difficult to understand because of the advanced mathematics involved in cryptography. However, you will see that there is a real simplicity to Bitcoin. Although the specifics of the mathematical properties are difficult to prove, it is not difficult to understand how they work. Therefore, we will look at cryptography and hashing, in particular.

8. Public Key Cryptography

Public Key Cryptography
Public key cryptography is a mechanism by which two people can exchange data publicly over the internet and generate a secret that only the participants know. The secrets are impossible to guess thanks to the large prime numbers that cryptography uses.

Public Key Cryptography

Inventors: Professors Whitfield Diffie and Martin Hellman (1976)
Prizes: The Turing Prize for Mathematics

Primitive Roots of Prime Numbers

Inventor: Johann Gauss (1801)

Diffie and Helman will go down in the annals of history with Pythagoras. Diffie and Helman invented public key exchange, which enabled users to generate a secret key by communicating public data. This invention was effectively the doorway to modern cryptography. It uses a property of prime numbers, modular maths, and something known as the discrete logarithm problem.

It is, in fact, simple. Modular maths works in the same as the remainder in division (positive numbers only).

For example, 12 ÷ 5 = 2 with a remainder of 2. 12 mod 5 = 2. It's literally that easy.

Now for Diffie Hellman. This is how two people generate a secret exchanging public data:

- We agree on a prime number* as our modulus. For example, we choose 17 and a generator number, 3. These numbers can be public.
- 3 mod 17 then becomes our 'source'.
- We each then choose a random private number. Let's say person A selects 15, and person B selects 13. A and B do not communicate their numbers to each other.
- Each person raises the generator to the power of their own secret number.
    - Person A: 3^{15} mod 17 = 6 (Check it in Excel, if you'd like, using the text '=MOD(3^6,17)=6')
    - Person B: 3^{13} mod 17 = 12
- Person A and Person B exchange these results with each other.

So far, so good. Here is the trick, the key to encryption, and the cornerstone of Bitcoin:

- Person A takes Person B's result, 12, and raises it to the power of their own private number.
    - Person A: 12^{15} mod 17 = 10
    - Person B: 6^{13} mod 17 = 10
- Both people get the same answer, *without knowing the other persons secret*. Magic.

With our small numbers above, we could, through trial and error, guess the secret numbers. With large primes, it is simply impossible because the number of possible solutions is incredibly large (more on that later). This property is known as the discrete logarithm problem.

That is encryption, and it relies on some superheroes of modern mathematics.

**The modulus needs to be prime (17, in our case) because one of the properties of prime numbers is the primitive root of a prime number (3 is a primitive root of 17). 3 raised to any power, however large, has an exactly equal chance of returning any integer between 0 and 17.*

9. Elliptic Curve Cryptography

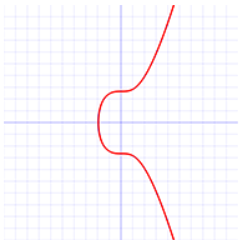
Elliptic Curve Cryptography (ECC)
ECC is the particular brand of cryptography used by Bitcoin. It is used to secure your bitcoins. There are 2^{256} possible combinations of a password available in ECC. That is more possibilities than atoms in the entire universe. For that reason, all the computing power in the world cannot hack into your Bitcoins wallet .

Fundamentally, it is cryptography that protects Bitcoin, and it is cryptography that keeps the whole system honest. In Bitcoin specifically, the encryption uses elliptic curve cryptography to secure its key pairs, and it, too, benefits from the discrete logarithm problem we explained above.

This is the formula for Bitcoin's elliptic curve:

Secp256k1 (Bitcoin) Elliptic Curve

$$y^2 = x^3 + 7$$



:

It works like this:

- We know the bitcoin ECC formula is $y^2 = x^3 + 7$
- Person A picks a secret number and generates a point on the elliptic curve using the formula.
- Person B picks a different secret number and does the same.
- A and B exchange their calculated points on the curve (over the internet—they never need to meet each other).
- Person A multiplies his secret number by person B's point, which generates a new point on the curve (due to the properties of ECC).
- Person B does the same and multiplies her secret number by person A's point on the curve.
- Once again, as per our simple example on the previous page, *they get the same answer—the same point on the curve.*
- This time, though, the numbers are much larger and generated by a more complex function.

Now, even if a third person knew all the data that had been exchanged by A and B over the internet, he would not be able to compute the new common point.

A and B never met, never spoke. However, they managed to generate a secret that cannot be hacked.

Elliptic Curve Cryptography

Inventors: Neal Koblitz and Victor S Miller (1985)

ECC is a weapons-grade security algorithm designed as a variant of typical public key cryptography algorithms.

ECC has advantages over RSA—the most widely used public key protocol—because of its significantly more efficient security relative to storage space.

Bitcoin's implementation of ECC is called Secp256k1.

10. Hashing

Hashing

Bitcoin uses hashing a lot. Hashing is just mixing up numbers and letters.

Let's say I hash the word 'bitcoin' by moving each letter down one in the alphabet:

b	i	t	c	o	i	n
+1	+1	+1	+1	+1	+1	+1
c	j	u	d	p	j	o

'Bitcoin' becomes 'cjudpjo'.

Hashing is as simple as that, but it normally has more steps.

Bitcoin also uses something known as hashing, which means exactly what it sounds like. Simply put, making a hash of a number or data set mixes up the input to produce a different output.

The SHA256 Hash

The particular form of the Bitcoin cryptography is known as a SHA256 Hash, and it has certain important properties.

- You always get the same output from the same input (known as deterministic hashing).
- It is impossible to generate the same output from different inputs, so each output is unique.
- The output is always 64 characters (256 bytes).

Do not underestimate the power of hashing. Messages of up to 2^{64} bit (2.3 exabytes, or 2.3 billion gigabytes) are transformed into single strings of data. For perspective, this means that an object seven times the size of Facebook's data warehouse in 2014 passed to SHA256 would produce a chunk of data the size of a 64-digit string and would uniquely represent that data.

The SHA256 Hash

Inventor: United States National Security Agency (NSA)

This simple tool undertakes SHA256 hashes:

https://passwordsgenerator.net/sha256-hash-generator/

It is even possible to compute them by hand. This excellent video shows you how:

https://youtu.be/y3dqhixzGVo

It works like our very simple hash above, except it involves moving data, summing in columns, and so on. Even so, it is simply the same steps repeated over and over.

An Example SHA256 Hash

Here, we change the first letter of the input from "l" to 1:

Input	Output
listedreserve	DA495CB857E65A13972D775A3486D5A6F16FFBA129E370A2885299C6C1849B2D
1istedreserve	F827992D29C736189A580512E1873F4C30EF5060B0E4D6E7C6AB2D1DE0998583

I changed one digit, and we got a completely different answer, and both answers are 64 characters long.

The Number of Possible Outputs

A further feature of the SHA256 hash is the sheer number of possible outputs it can generate. In fact, it is a calculable number; 2^{256}. This number is massive—beyond the number of atoms in the universe. Again, this property is important, as well shall see.

2^{256}

The number is an important one in Bitcoin. Through the protocols use of the SHA256 hash, it represents the maximum mining difficulty and the maximum number of potential public and private keys available.

It is estimated there are between 10^{78} and 10^{82} atoms in the universe. You can see the sort of numbers Bitcoin uses is large beyond what a human can perceive, 2^{256} being many orders of magnitude larger than the universes atom count

We now have the tools we need to understand Bitcoin further.

11. Bitcoin Mining: A Puzzle

Bitcoin Mining

Bitcoin mining is a competition to solve a cryptographic puzzle, and it serves two purposes:

1. To check transactions every ten minutes to ensure that they are valid

2. To mine new bitcoins, growing the supply to 21 million coins

The successful miner will select only the valid transactions and publish them to the network, as well as collect 12.5 bitcoins.

Mining is a huge business. Each block is worth US$150,000.

Bitcoin mining is simply computers attempting to solve a cryptographic puzzle. The puzzle has 2^{256} possible answers. No amount of computing power could solve this puzzle, so the Bitcoin protocol sets the puzzle in a unique way that requires an answer in a given range. The range is adjusted so that it takes ten minutes to solve. The more computers that try to solve the puzzle, the smaller the range, and the ten-minute window is preserved.

How to Mine a Block

Users create the puzzles by including the following information in what is known as a candidate block (if you solve the puzzle, it becomes a real block):

i) Software version number
ii) Previous blocks hash
iii) Merkle root of the selected transactions (a hash of the root of the transactions selected for the block)
iv) Timestamp (seconds from the Unix epoch)
v) Difficulty target (proof of work difficulty target for this block)
vi) Nonce

We will deal with each piece of information in turn, but they look a bit like this (real block data below).

version	02000000
previous block hash (reversed)	17975b97c18ed1f7e255adf297599b55 330edab87803c8170100000000000000
Merkle root (reversed)	8a97295a2747b4f1a0b3948df3990344 c0e19fa6b2b92b3a19c8e6badc141787
timestamp	358b0553
bits	535f0119
nonce	48750833

i) Software Version Number

The software version number is relevant because from Bitcoin upgrades the software from time to time. The version number is used to track upgraded software's adoption rate.

Note that Bitcoin is backwards compatible, so if you are using the version of the software from 2009, it will still work.

ii) Previous Block Hash

Each block has a unique hash that is simply a function of the bullets above. Therefore, every block contains the previous block's hash—this is the source of the word 'blockchain'.

Blockchain

Inventor: Satoshi Nakamoto 2009
Prizes: none

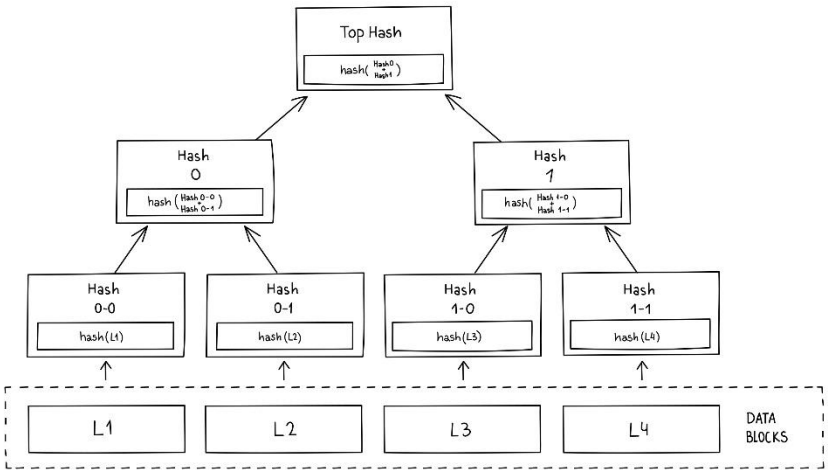
'Blockchain' is an overused term. It arises simply because every block contains the hash of its predecessor. As a result of the properties of the hashes, all blocks are then tied to one another, and any change to the history changes everything—hence, 'blockchain' and the word 'immutability'.

Blocks are chained together because they each include a unique value generated by their predecessors. If anyone changes the history, remember the SHA256 hash—if you change even a single digit, everything changes.

Each block contains a hash of the previous block, which itself contains a hash of its predecessors. Any change to any of them collapses the whole chain because everything would change, and the whole network would reject the change.

iii) Merkle Root

Merkle trees are named after Ralph Merkle, one of the co-inventors of public key cryptography. Merkle trees are another piece of genius that makes up Bitcoin. They allow large data sets to be summarised as single strings of numbers as follows using a hash:



Let's say I have four transactions: L1, L2, L3, and L4. I hash those transactions using the same process as above. I then pair them and hash them. I keep going up the tree until I have one hash left. This hash is known as the Merkle root. It represents all the transactions in the block. I cannot generate the same root unless I use the same transactions, which means that I can summarise an entire block of data in one string.

Note the transactions must be paired. For an odd number of transactions, the protocol simply hashes the last transaction twice.

Merkle Trees

Inventors: Professor Ralph Merkle (1979)
Prizes: IEEE Richard W. Hamming Medal (2010)

By combining two transactions in a hash, I can reduce the size of those transactions by half. I can continue doing this with any even number of transactions until I am left with one. This last transaction is known as the Merkle root.

Bitcoin transactions are summarised in blocks exactly like the Merkle root.

Merkle trees have amazing properties. The Merkle root is unique to the specific transactions it combines. Furthermore, given the Merkle root, I can confirm that a transaction is included in that root quite easily without needing to unravel the whole thing (which would take years). This property is very special and useful.

iv) Timestamp

The timestamp uses the Unix epoch—that is, the number of seconds that have elapsed since 1 January 1970. It provides a convenient single figure for time. The Unix time as of writing this sentence is 1562566408. Once again, it is easy to convert if you want to try: https://www.unixtimestamp.com/.

v) Difficulty Target

The difficulty target is simply how hard the puzzle is to solve. Bitcoin is designed so that it takes ten minutes to produce a block (an arbitrary choice that trades efficiency for stability). The difficulty target adjusts depending on the total amount of computing power attempting to solve it so that probabilistically, someone will find an answer within the difficulty range in ten minutes.

Rather than adjust by time specifically, Bitcoin adjusts by block count. In a two-week period, we expect

- 6 blocks per hour
- 24 × 6 = 144 blocks per day
- 14 × 144 = 2,016 blocks in two weeks

However, if miners average nine minutes per block because there is a lot of new hash power, then 2,016 blocks will be produced in 12.6 days (9 ÷ 10 × 14). Therefore, the difficulty adjustment will increase in the correct ratio so that the next 2,016 blocks take 14 days.

The candidate block we are creating must include the current difficulty so that every other participant knows you tried to solve a puzzle as hard as they did.

The difficulty adjustment itself is genius. It time-proofs Bitcoin against advances in computing power. Every miner must publish the difficulty they used in their candidate block. If you try to cheat, everyone will know, and all your mining effort will be wasted.

The Difficulty Adjustment

Inventor: Satoshi Nakamoto 2009
Prizes: none

Every two weeks, the mining difficulty is adjusted by the Bitcoin protocol. The difficult adjustment shows real foresight:

- It correctly anticipates and accommodates Moore's Law of computing power.
- It correctly anticipated that if Bitcoin were successful, huge amounts of hash power would be thrown at the network.
- The network now has more computing power than any other computing network in the world—currently 7.08[19] hashes per second. For context, that is a number of computations equivalent to the number of grains of sand on Earth.
- Blocks are still taking ten minutes to produce, thanks to the difficulty adjustment.

vi) The Nonce

A nonce is simply a random number that completes the block header. The nonce is set to 0 initially.

Candidate Block

I constructed my candidate block, which is simply a string of numerical values as outlined above. Here is an example of a mined block (note the hash is not 0 because this is a mined block, not a candidate):

version	02000000
previous block hash (reversed)	17975b97c18ed1f7e255adf297599b55 330edab87803c8170100000000000000
Merkle root (reversed)	8a97295a2747b4f1a0b3948df3990344 c0e19fa6b2b92b3a19c8e6badc141787
timestamp	358b0553
bits	535f0119
nonce	48750833

You can see from the number of blue boxes in the mining section, that mining is at the heart of Bitcoin. Many of the Bitcoin's inventions are involved here.

12. Solving the Mining Puzzle

Successfully Mining

A successfully mined block

- Produces a valid answer to the puzzle that everyone can check
- Contains only valid transactions (checked by the network before the block is accepted)
- Earns the miner 12.5 bitcoins + fees

I now have the candidate string that I generated earlier. Mining is simply the following:

- Hashing the string I created earlier with a SHA256 hash
- Determining if the answer is less than a target value set by the difficulty level (see below)
- If no, then changing the nonce and trying again
- Repeating until my answer is below the target or until some other miner gets an acceptable answer

Mining doesn't have one answer. I just need any answer below a certain value—the target value. Remember, there are 2^{256} possible answers, which as we know, is large. Therefore, the probability of hitting the correct hash is $1 \div 2^{256}$. You could mine from the beginning of the universe and not get the correct answer.

Bitcoin only requires that you get an answer below a certain target. As the difficulty rises, the target range gets smaller. Likewise, when the difficulty is lower, the target range is bigger. Think of it like archery: as the difficulty rises, the target gets smaller. That is what is happening here—the target difficulty is simply setting the probability of finding a correct answer at a level that will take ten minutes based on the current mining power available to the network.

As you can imagine, all this work is all highly processor intensive. It uses 100% of a CPU's power and can consume CPU power without limit.

13. The "Answer"

Block times can vary a lot, and they will be ten minutes on average, but in reality, they are anywhere from 30 seconds to half an hour.

Let's say that after eight minutes, someone finds the answer. The successful miner then publishes his block with the answer. His hash will be reported to everyone else on the network who will verify the following:

- He used the correct difficulty "Proof of Work" (he tried to solve a problem as hard as everyone else did)
- The transactions in the block are correctly signed (with the right private key – see part 2)
- The block is no greater than the specified size
- There are no transactions in the block that have been spent before (our double-spending problem)

This validation is done very quickly. Remember the properties of encryption—when I have the answer, it is trivial to confirm it is correct even though the answer was unbelievably hard to find. It's a one -way function, and in seconds, we are on to the next block.

Provided the criteria are met, the network accepts the block, and the miner mints his own reward: a Coinbase transaction where he awards himself 12.5 bitcoins, which is currently US $150,000. The process now repeats and has repeated just like this, completely uninterrupted, for ten years.

14. Proof of Work

Proof of Work

You cannot forge a Bitcoin block. You must expend a lot of energy to create one, and it is expensive to do. If you attempt a forgery, it will be rejected by the network, and all the expense of your forgery effort will be wasted.

Proof of work is commonly referred to in Bitcoin. It simply means that I have expended energy through my use of computing power, determined by the difficulty agreed for that block. It is fundamental to the value of bitcoin because the energy consumed in the consumption of the block is key to the value of Bitcoin.

The current amount of work to hash a Bitcoin block is 70,000,000 TH/s—that is, 70 million trillion calculations per second. That is more calculations per second every second than there are grains of sand on earth. The point is, it is already an enormous business with more computing power than Facebook, Google, and Amazon could muster combined.

One of the reasons that Bitcoin is considered 'hard money' is because of the proof of work built up in the network. We will consider this concept in more detail in a later paper.

Proof of Work

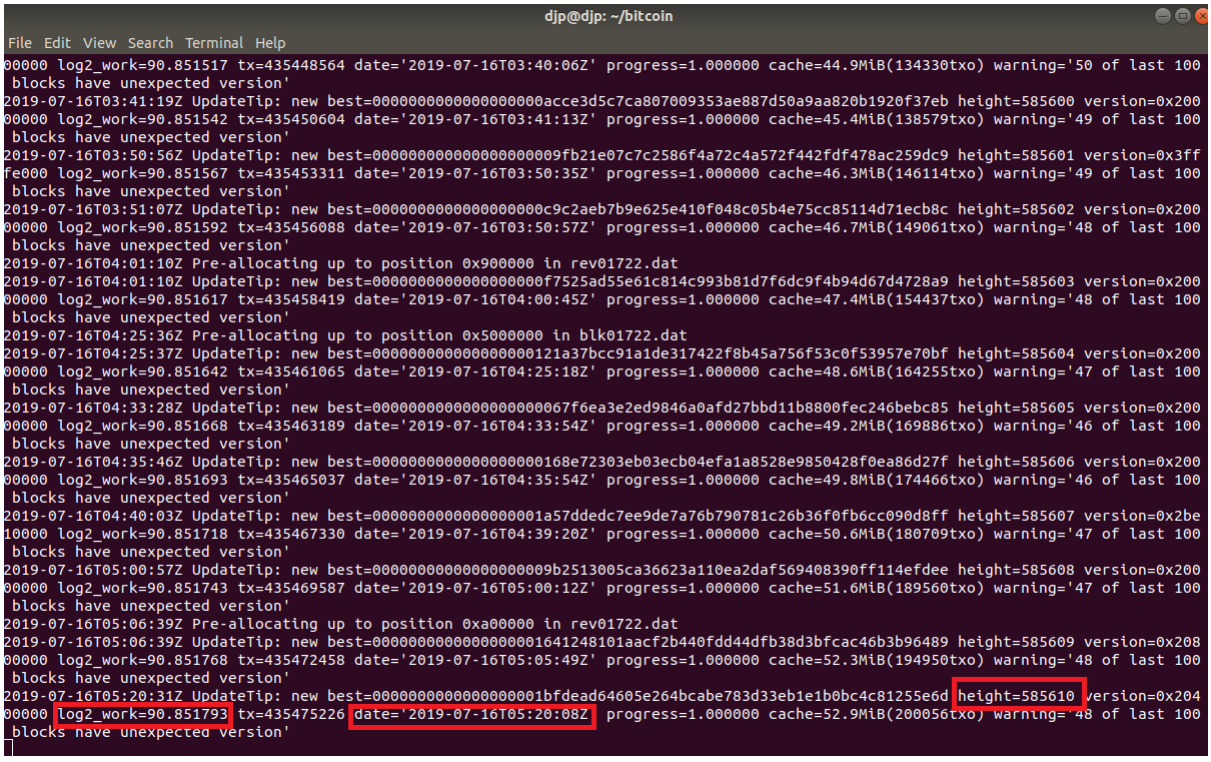
Inventor: Dr Adam Back 1997
Prizes: none

Proof of work was first invented as an email spam prevention tool by British cryptographer Dr Adam Back. He designed a system that required a small amount of CPU work from your computer before you could send an email. To send one email would be trivial, and you wouldn't notice. Send 100, and your computer would slow down for a while. Send 10,000 (spam), and you would lock your computer for hours or days before the emails sent.

The idea was that emails would include a hash stamp that would prove you had spent computing effort to send the email. Anyone sending spam wouldn't go to the effort because the cost would be too high. Simply put, this stamp would make the cost of sending spam higher than the benefit it might yield.

15. What Does Bitcoin Look Like?

Below is a screenshot of a live blockchain. It prints a new line with each block.



You can see a couple of interesting things by looking at the blockchain, notably

- Block height: 585610 (the current block we are mining)
- Log2_work 90.851793:
  - This is the cumulative proof of work in the blockchain and represents the total number of calculations that would be required to replicate where it is today
  - It means $2^{90.851793}$.
- Date: 16 July 2019 at 05:20 a.m.

16. Conclusion of Part 1

As you can see, Bitcoin uses some extremely powerful pieces of mathematics to generate its inputs and outputs and protect its ledger with encryption. These properties protect Bitcoin from attack, and ten years of uninterrupted operation proves it.

You may hear the question, 'Who invented Bitcoin?' The point we are making throughout this paper is that the number theory and special properties of Bitcoin have been under development since the 1800s, starting with Gauss (the greatest mathematician of the last 500 years), to the development of encryption in the 1970s and 1980s by Diffie and Hellman (who won the Turing Prize for their amazing work), followed by Adam Back's proof of work in the 1990s.

Bitcoin has been a long, long time in the making. It is epoch-making technology, just as important as double-entry bookkeeping was to the development of modern business and trade. Up until Bitcoin, property rights were a social construct, reliant on the law to be enforced. Now we can rely on mathematics. That so many people have yet to get their heads around it only proves the point.

17. What Next?

In the next instalment, we will cover

- Coinbase transactions and the money supply
- Bitcoin wallets: private and public keys
- Sending transactions: how is it done?
- Deterministic wallets