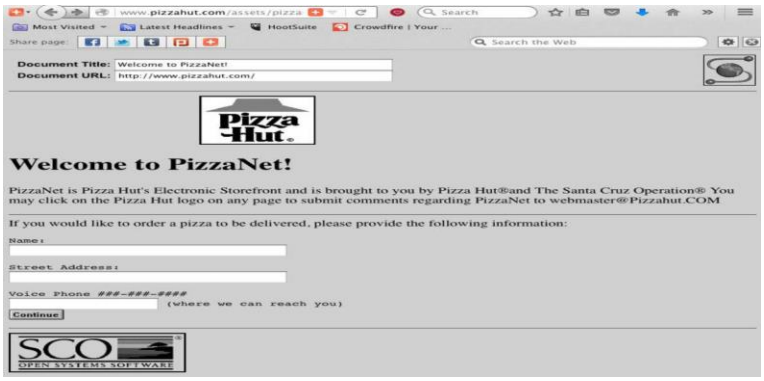
Blockchain Development Independent Study

12/9/2021

Blockchain Background & History

The first online sale: Pizza Hut - 1994, though money changed hands upon delivery.



Problem discovered in the early days of the web when money was beginning to be exchanged through services such as eBay, Amazon, Pizza Hut, etc.: How to digitally move value peer-to-peer without any trusted central intermediary (Gensler).

Early cryptographic digital currencies failed: DigiCash – 1989, CyberCash – 1994, Hashcash – 1997, Bit Gold – 1998. Why: Lack of adoption, centralization, double spending, consensus issues. Digital payment innovation continued through PayPal – 1998, AliPay – 2003, etc. The internet was built on decentralized protocols (TCP/IP, HTTP, SSL/TLS, etc.), however payments still relied on the traditional credit card/banking system.

From: Satoshi Nakamoto <satoshi <at> vistomail.com>
Subject: Bitcoin P2P e-cash paper
Newsgroups: gmane.comp.encryption.general
Date: Friday 31st October 2008 18:10:00 UTC
“I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party.” (Satoshi Nakamoto Institute)

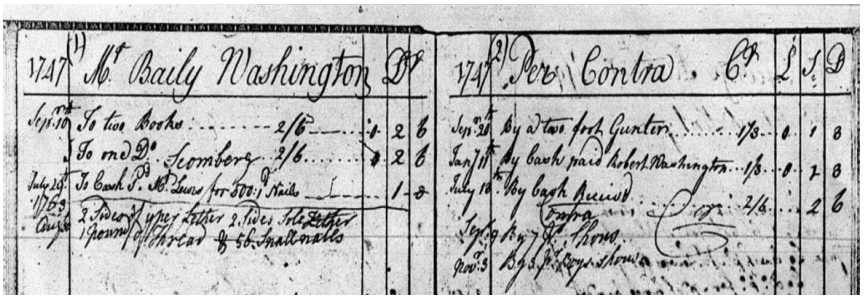
This was the first message that the anonymous person(s), who went by the name of ‘Satoshi Nakamoto’ sent. It was sent to an internet forum of other cryptographers and attached was the whitepaper for Bitcoin, which outlined the societal need for a peer-to-peer online payment system that was self-governing, secure, and limited in quantity.

Bitcoin was built using the “new” technology called blockchain. “A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography.” - Wikipedia

Original concept introduced, 1991, paper titled "How to time-stamp a digital document", authored by Stuart Haber and W. Scott Stornetta, it was meant to show how to prevent time-stamped documents from being back/forward dated. While the word blockchain wasn't the paper, most of the ideas and features of blockchain were present.

Blockchain is frequently labeled as being a 'distributed ledger', but what does this mean?

A ledger is a collection of accounts in which account transactions are recorded. Each account has an opening or carry-forward balance and record transactions as either a debit or credit in separate columns. This allows the **authority in charge of the ledger** to keep track of the balances of all accounts. A modern example of this being Government Ledgers of who owns what property (when property is traded between properties, this is recorded on the ledger).

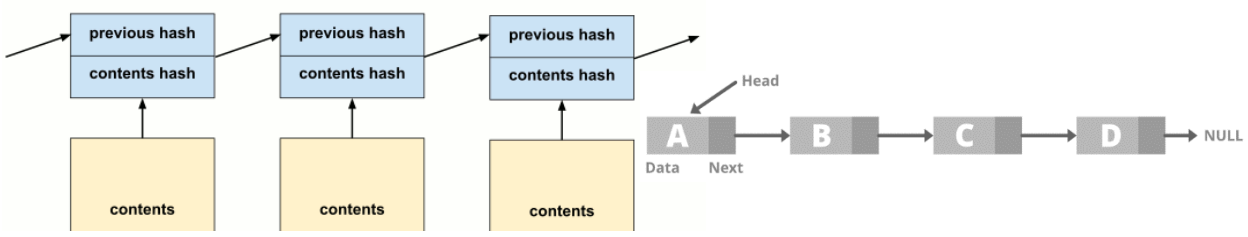


George Washington's personal ledger – 1714

A ledger which is said to be distributed, such as blockchain, is not kept controlled by one central authority, but instead by multiple different authorities who collectively maintain control. In Bitcoin's case, any modern computer is able to self-proclaim itself as one of these authorities and assist in the process maintaining the validity of the ledger. (Source)

Fundamentals & Cryptography

The "Blocks" that make up a blockchain consist of data (transactions, smart contracts, etc.), a previous hash, which is the same hash as the block's predecessor, and its own hash, which is created from running a hashing algorithm on its own data.



Simplified blockchain illustration (left), conceptually similar to singly linked list (right).

The difference between a blockchain and a singly linked lists is that a blockchain is immutable, meaning that, similar to a ledger written in pen, it's not possible to change its existing entries. This concept be explored further looking into how hashing works.

An essential component of blocks which is used to ensure their validity as part of the overall blockchain is a hash function. As shown in the previous figure, this hash is the linking factor between each of the blocks. Many types of hashing algorithms exist, some of the more commonly used ones include MD-5, SHA-3, and SHA-256 (Hashing Algorithms). These hashing algorithms deviate from each other in both their output sizes and from how they work under the hood, however they all adhere to these five hashing algorithm general requirements:

- One-Way
  - It's not possible to restore / reverse engineer the input data from the resulting hash
- Deterministic
  - When the hashing algorithm is applied multiple times on the same input data, the resulting hash will always be the same
  - The resulting hash for input data of any size will always be the same size
- Fast Computation
  - It should be computationally efficient to compute; O(n) or better
- Avalanche Effect
  - Making a slight change (as small as one bit) to the input data will result in a significantly different resulting hash
  - Called 'avalanche' because in the algorithm, small changes trigger other changes, which in turn trigger more changes, etc.
- Withstands Collisions
  - There can't be any way which malicious actors are able to generate data which, when passed through the hashing algorithm, will produce a matching hash of the hash of other known input data
  - Infeasible to find and x and y where Hash(x) = Hash(y)

SHA-256 Hashing Algorithm Demo

1. Type any word in the box to demo the associated hash being calculated (efficiently)
2. Add/Remove a character from the word to demo the avalanche effect
3. Point out the first 3 characters of the resulting hash (Ex. "Hello" -> 185)
4. Clear the box and type the same word as before again, point out the deterministic property of the hashing algorithm, as the first 3 characters are the same (185)
5. Paste a large text block of text in the box, such as the constitution to demo that the resulting hash size is always 256 bits, regardless of the input size
6. Optionally, add text representing a transaction ("Bob 10 BTC -> Dan")

Now that hashing algorithms are understood, the immutability of blockchain can be further explained. In the illustration of a blockchain below, the blocks are connected since the **hash value** of each previous block matches the recorded **previous block hash value** stored in each of the blocks. The blocks are cryptographically linked, or "chained", together through this connection.

When a new block is given permission to be added to the block chain, it's 'previous block hash' will match the hash of its preceding block.

For examples sake, it could be said that the data that each of these blocks encompass is financial transactions between different accounts. If a malicious actor were to go back to a previous block and change a transaction such that they would benefit from it, this would change the hash of the block when recalculated. This is because the hash is comprised of both the data in the block and the previous block's hash.

The result of this blocks hash being changed would then be able to be easily detected by an agent when it's verifying that all previous hash block values are matching the hashes of the previous blocks. This concept why the immutablity of blockchain is gaurnteed. (NDC)

Mining & Proof-of-Work

It's been established that blocks are linked together through their hashes, and that the input data to create a blocks hash consists of all the data contained within the block, and the previous blocks hash. The question might be asked, if it's as simple as linking different blocks of data by hash, and hashing is efficient, "what's the deal with mining"? Or, "why are so many resources being used to 'mine' new blocks, more specifically, for Bitcoin"?

To answer those questions, an additional element of blocks, contained within their data, must be introduced. This element is called a **Nonce**, which is defined as 'occurring, used, or made only once or for a special occasion' (Webster).

'Mining' new blocks is simply the act of continually calculating the hash of the combined value of the existing block data (timestamp, previous hash, transactions) and an arbitrarily chosen nonce value, until a hash is calculated which matches a certain pattern. Finding this pattern is referred to as the **Difficulty** to mine a block, and it varies over time (Difficulty). The pattern associated with Bitcoin's difficulty is defined by the leading amount of 0's in a block's hash.

The amount of leading 0's required for the block's hash is algorithmically adjusted such that a new block is mined approximately every 10 minutes. The amount is adjusted every 2016 blocks (about every two weeks), and is currently 19 leading 0's. (Source)

Block 713327 (12/9/21) – 19 leading zeros:

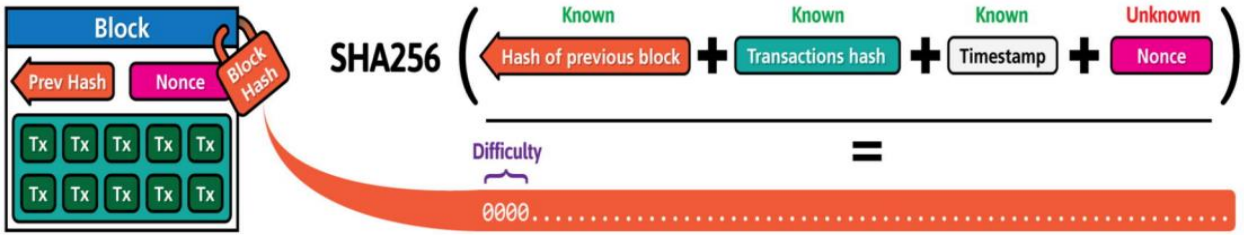
00000000000000000009b67c39861b803427115db0d4cef19c0bd39440d01266

Genesis Block (1/3/09) – 10 leading zeros, though only required 8:

000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

(Bitcoin Explorer)

To add a block to a blockchain, computers from around the world are trying to mine blocks by generating over 89 quintillion hashes every second by changing nonces (& rearranging transactions). Once a nonce is found by one of these computers which results in the generation of a hash matching the difficulty pattern, the nonce is communicated to the rest of the network, and efficiently verified, the computer who found the nonce is rewarded, and the new block is added to the blockchain. In this case, the consensus between all the nodes of the network to add the new block is called **Proof-of-Work** because evidence of work, through expending computational power to calculate nonces, is required to achieve consensus.



Simple Block & Blockchain Demo

1. Show that the box on screen represents an individual block, not linked to any others. Mention that all the information within the block (Block #, Nonce, Data) is the input to the hashing algorithm which, in this case, is SHA256
2. Point out that in this case, the difficulty of mining blocks is finding nonces with at least 4 leading zeros, the current nonce (72608) was selected to produce the current hash that begins with 4 leading zeros
3. Demo the avalanche effect completely changes the hash by typing any text in the data field, note how the block is now red because its hash doesn't match the difficulty pattern (4 leading zeros)
4. Then, click the mine button, behind the scenes this will iterate the nonce until it finds one which creates a hash that matches the difficulty pattern
5. Next, move to the 'Blockchain' tab at the top of the page

6. Explain that each of these blocks are nearly identical to the block on the previous page, however, now they take the hash of the previous block into account when generating their own hash
7. In each of the first few blocks type some text in the data fields (this text could be formatted like transactions for maximum effect)
8. Next, starting from the first block (the genesis block), mine each of the blocks that text was added to
9. Finally, change the data in one of the blocks in the middle of the chain, and point out how this invalidates all blocks to the right of said block, as the hashes are now mismatched, mention once again that this is what makes blockchains immutable

The Evolution of Bitcoin Mining: (Coindesk)

- Mining was originally done on CPUs from 2009 to 2010, capable of 2-20 MH/s
- Next it was discovered that hashing could be parallelized, so from 2010 to 2013 GPUs were used, capable of 20-300 MH/s
- Finally, Application Specific Integrated Circuits (ASICs) were developed specifically for calculating the SHA256 hash. Almost the only thing these computers can do is mine bitcoin, therefore they're many orders of magnitude greater at mining when compared to GPUs, they are capable of 4-16 TH/s

->


->



Modern Mining Factory

Quantum Vulnerabilities & Blockchain Security

Despite the field of quantum computing currently being in its infancy, in the past few years it's been experiencing growth, and along with this growth, the fear of technologies which use modern cryptography being rendered obsolete also grows. (Forbes) Blockchain is one of these technologies, using cryptography to both encrypt transactions and to link blocks.

One discovered use case for quantum computers is their ability to factor large numbers down to two component primes using Shor's Algorithm. (Microsoft) The inability of classic computers to do this factorization in an efficient manner is the fundamental principle modern asymmetric key encryption algorithms such as RSA and ECDSA. These encryption algorithms are used for the keys for wallets (which are used to send transactions) for most modern blockchains such as Bitcoin and Ethereum (Nakamoto). If quantum computers continue along their growth trajectory and modern encryption algorithms stay in use, the security of most blockchains are placed at risk.

Fortunately, all hope is not lost for blockchains. The US National Institute of Standards and Technology is currently gathering proposals for "post-quantum" encryption algorithms which are theoretically secure against even the largest of future quantum computers. (NIST) They're doing this because they estimate that large enough quantum computers to disrupt classical encryption will arrive in the next twenty years. A soft fork can be used for blockchains to replace their current encryption algorithms with the newly proposed post-quantum algorithms to prevent quantum vulnerabilities. (Fork)

In terms of linking blocks, modern hashing algorithms are theorized to be quantum resistant to detect collisions. (Cryptology) However, more research is currently being done to mathematically confirm this. Currently, one of the most efficient theoretical implementations of a quantum computer to detect a SHA-256 collision is less efficient than the classical implementation for breaking the standard. (GitHub)

Cryptocurrencies & Alternative Consensus Mechanisms

TODO: Discuss native currency

Pos – Nodes validate next block according to random selection based on stake in native currency, TODO: add more detail

DPoS – TODO: read this article https://www.gemini.com/cryptopedia/proof-of-stake-delegated-pos-dpos#section-history-of-po-s

PPoS – TODO: read this article https://community.algorand.org/blog/understanding-the-technicalities-behind-the-pure-proof-of-stake-protocol-algorand-uses/#:~:text=Unlike%20the%20Bonded%20Proof%2Dof,ability%20to%20spend%20their%20stake.