NITDA

FEDERAL MINISTRY OF COMMUNICATIONS AND DIGITAL ECONOMY

Leveraging ICT for National Development

Introduction to Bitcoin Theory Part 1

DOMINEUM

BSV BLOCKCHAIN

Calistus Igwilo
https://linkedin.com/in/calistus-igwilo
https://twitter.com/CalistusIgwilo

Abstract

A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Peer to Peer Cash

A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution.

- Satoshi Nakamoto, Bitcoin Whitepaper

Peer to Peer Cash

Role of Traditional Financial Institution

- Central banks distribute currency and establish money-related policies

- Commercial banks accept deposits and offer business and consumer loans
    - Secondary functions:
        - transfer of funds
        - periodic collections and payments
        - letters of credit
        - foreign exchange, etc.

- They are established and regulated by the law

Peer to Peer Cash

In Bitcoin, money is exchanged peer-to-peer using the Bitcoin protocol.

Peer to Peer Cash

Wallets allow users to create transactions that sign ownership records of digital coins and assign them to new owners.

Programatically, it refers to the data structure used to store and manage a user's keys

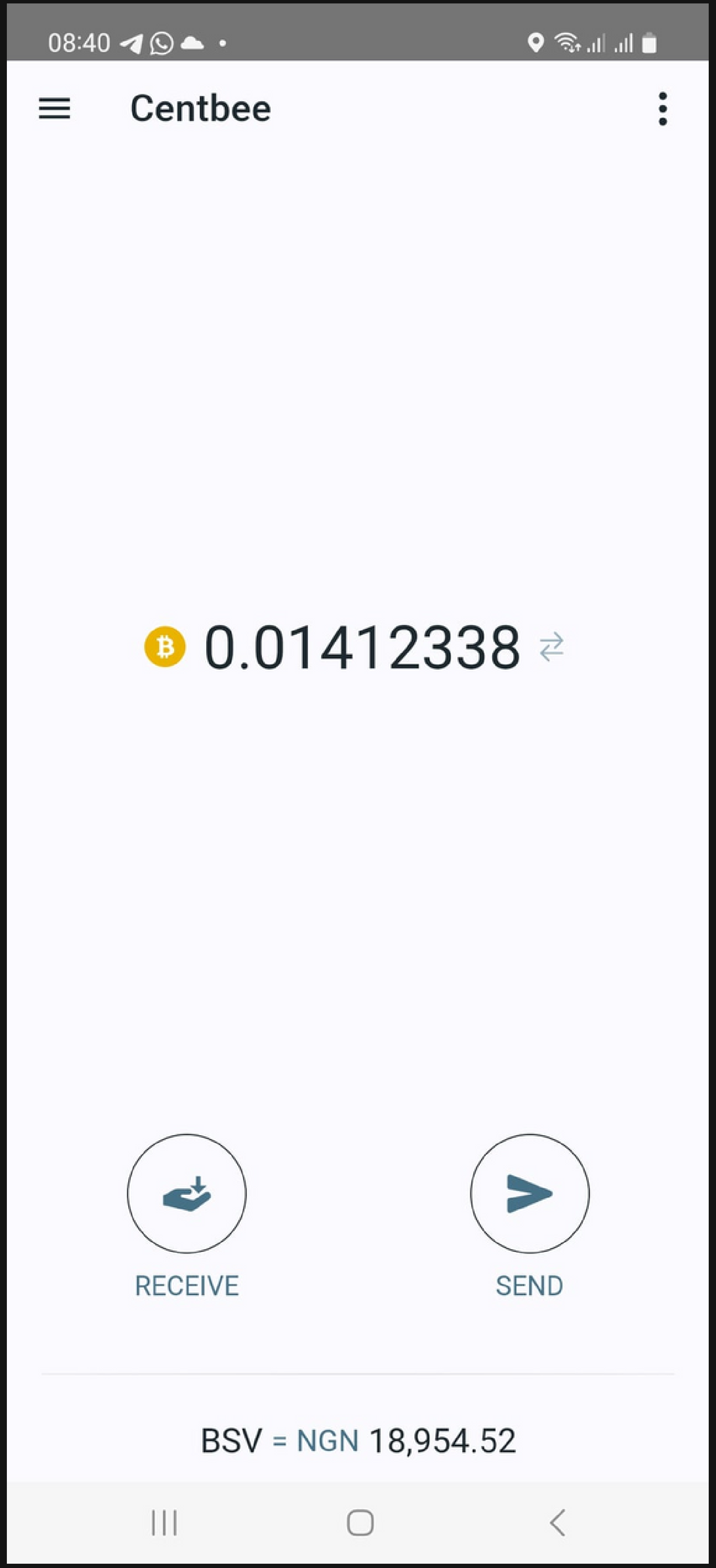
Types of Wallets

- Desktop full node wallets
- Mobile lightweight wallets
- Web third-party wallets
- Hardware wallets

Peer to Peer Cash

Types of Wallets

- Desktop full node wallets
- Mobile lightweight wallets
- Web third-party wallets
- Hardware wallets



Peer to Peer Cash

Mneumonic Words

- 12-24 English words, selected randomly by wallet software

- Used as the basis for the keys that are generated by the wallet

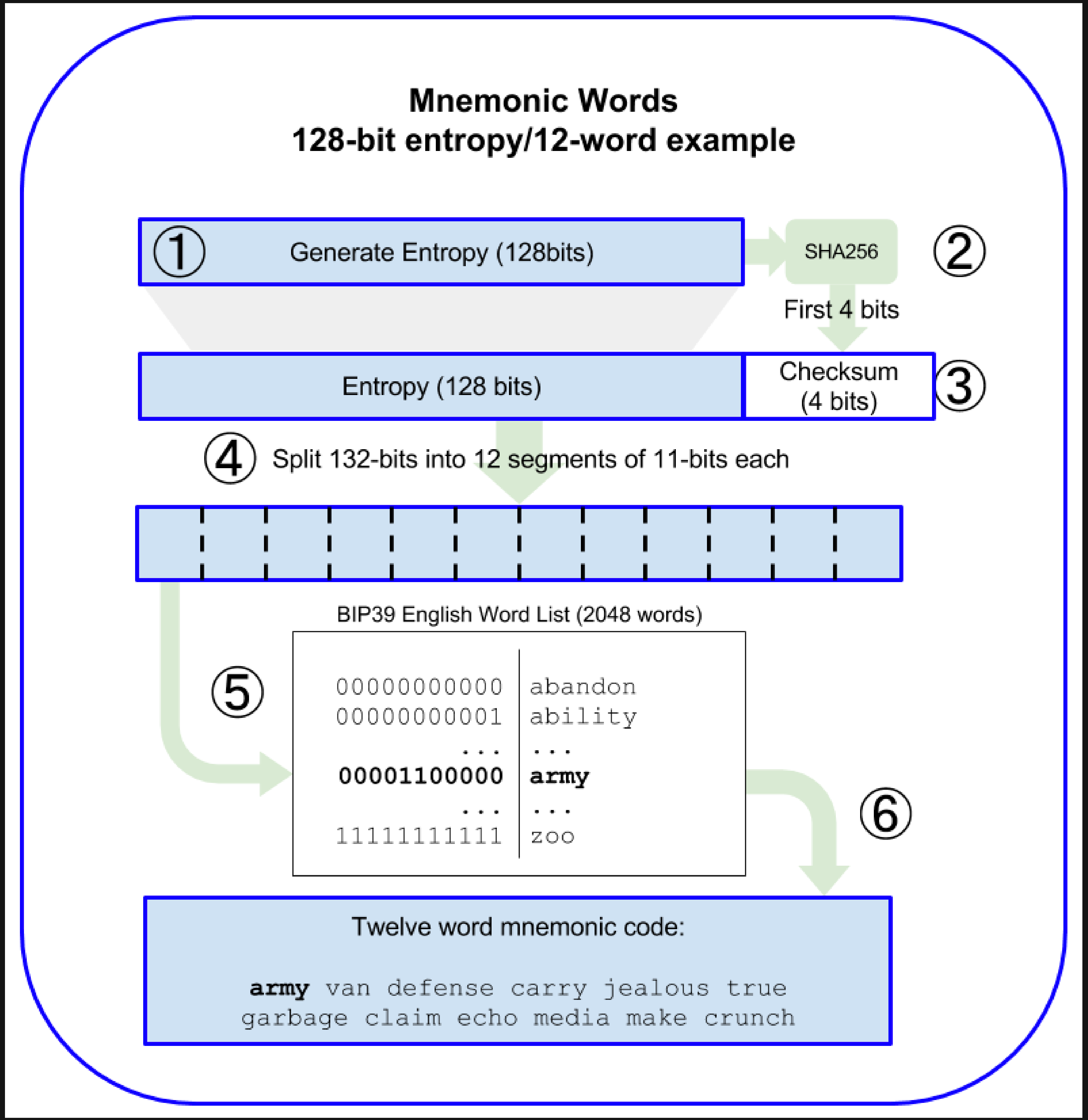
- Used to restore all the transactions and digital assets in a wallet

Peer to Peer Cash

Mneumonic Words Generation

The wallet starts from

- a source of entropy
- adds a checksum
- maps the entropy to a word list


Mnemonic Words
128-bit entropy/12-word example
①
Generate Entropy (128bits)
SHA256
②
First 4 bits
Entropy (128 bits)
Checksum (4 bits)
③
④
Split 132-bits into 12 segments of 11-bits each
BIP39 English Word List (2048 words)
⑤
00000000000 abandon
00000000001 ability
... ...
00001100000 army
... ...
11111111111 zoo
⑥
Twelve word mnemonic code:
army van defense carry jealous true garbage claim echo media make crunch

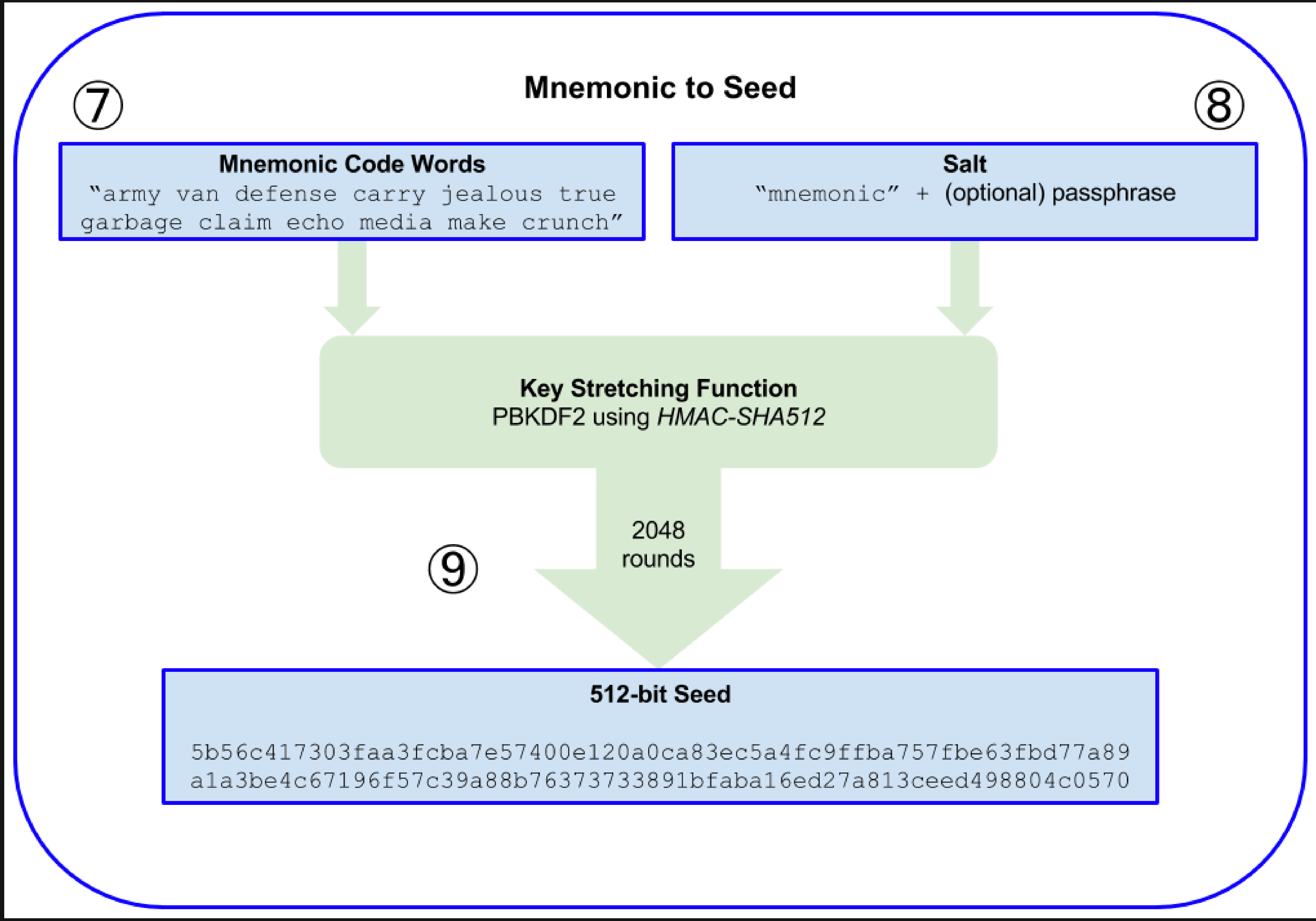
Peer to Peer Cash

Generate seed from Mneumonic

- The mnemonic words represent entropy with a length of 128 to 256 bits.

- The entropy is then used to derive a longer (512-bit) seed through the use of the key-stretching function PBKDF2.

- The seed produced is then used to build a deterministic wallet and derive its keys.


Mnemonic to Seed
⑦
⑧
Mnemonic Code Words
"army van defense carry jealous true garbage claim echo media make crunch"
Salt
"mnemonic" + (optional) passphrase
Key Stretching Function
PBKDF2 using HMAC-SHA512
2048 rounds
⑨
512-bit Seed
5b56c417303faa3fcba7e57400e120a0ca83ec5a4fc9ffba757fbe63fbd77a89
a1a3be4c67196f57c39a88b76373733891bfaba16ed27a813ceed498804c0570

Peer to Peer Cash

Storing Mneumonics Safely

- If not protected enough, the mnemonic will be at risk of being stolen.
- If protected too much, mnemonic will be at risk of being permanently lost.
- Balance the risks by writing two copies of the mnemonic phrase on paper, with each of the words numbered as the order matters.
- Store each copy in a separate secure location such as a locked desk drawer or a fireproof safe

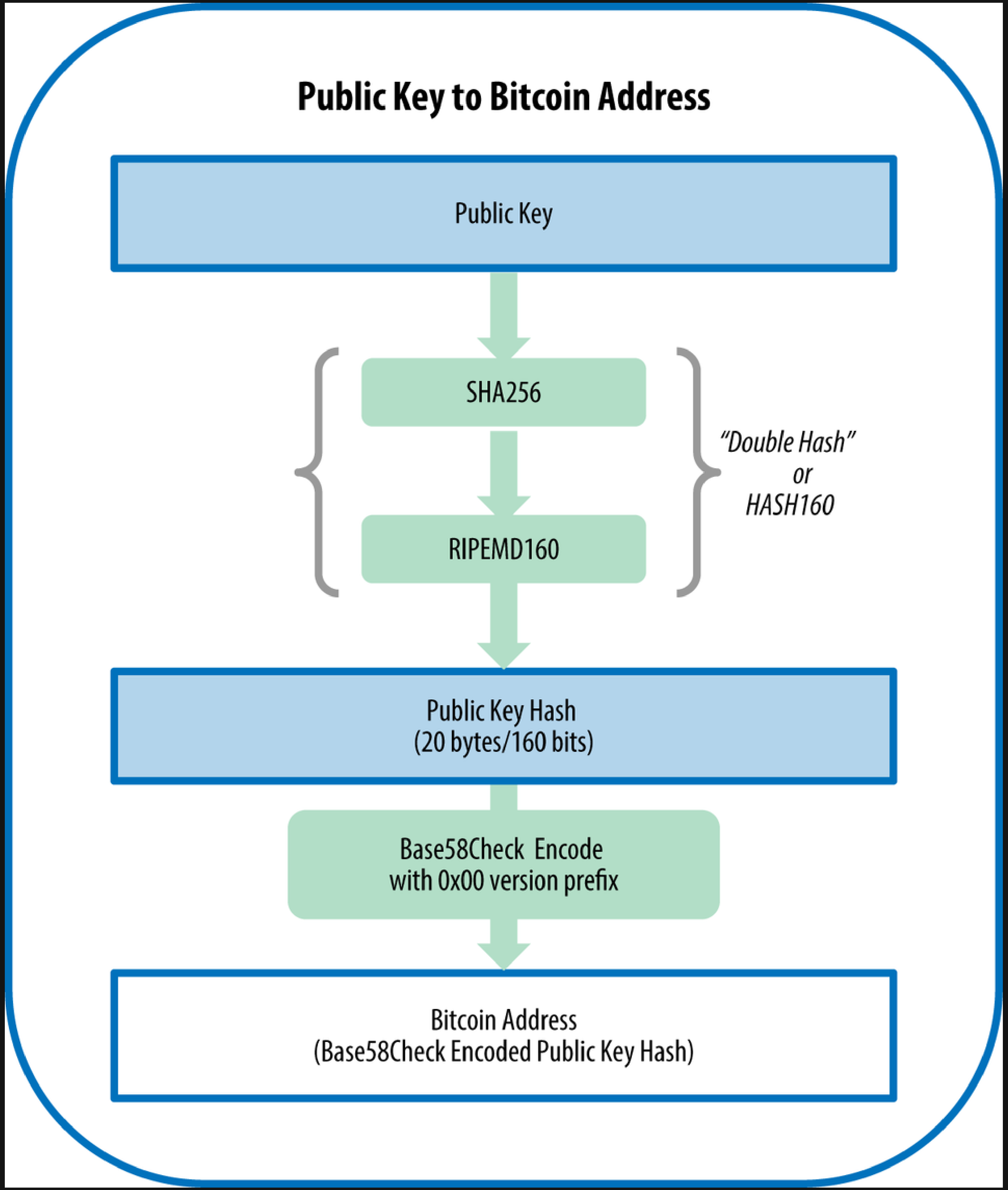
Peer to Peer Cash

Digital Keys

- Keys come in pairs: private (secret) key and public key
- Keys can be generated and managed by the user's wallet software without reference to the blockchain or access to the internet
- The public key is used to receive funds, and the private key is used to sign transactions to spend the funds
- Bitcoin addresses can be generated from public key
    - Some addresses can represent a script, though.

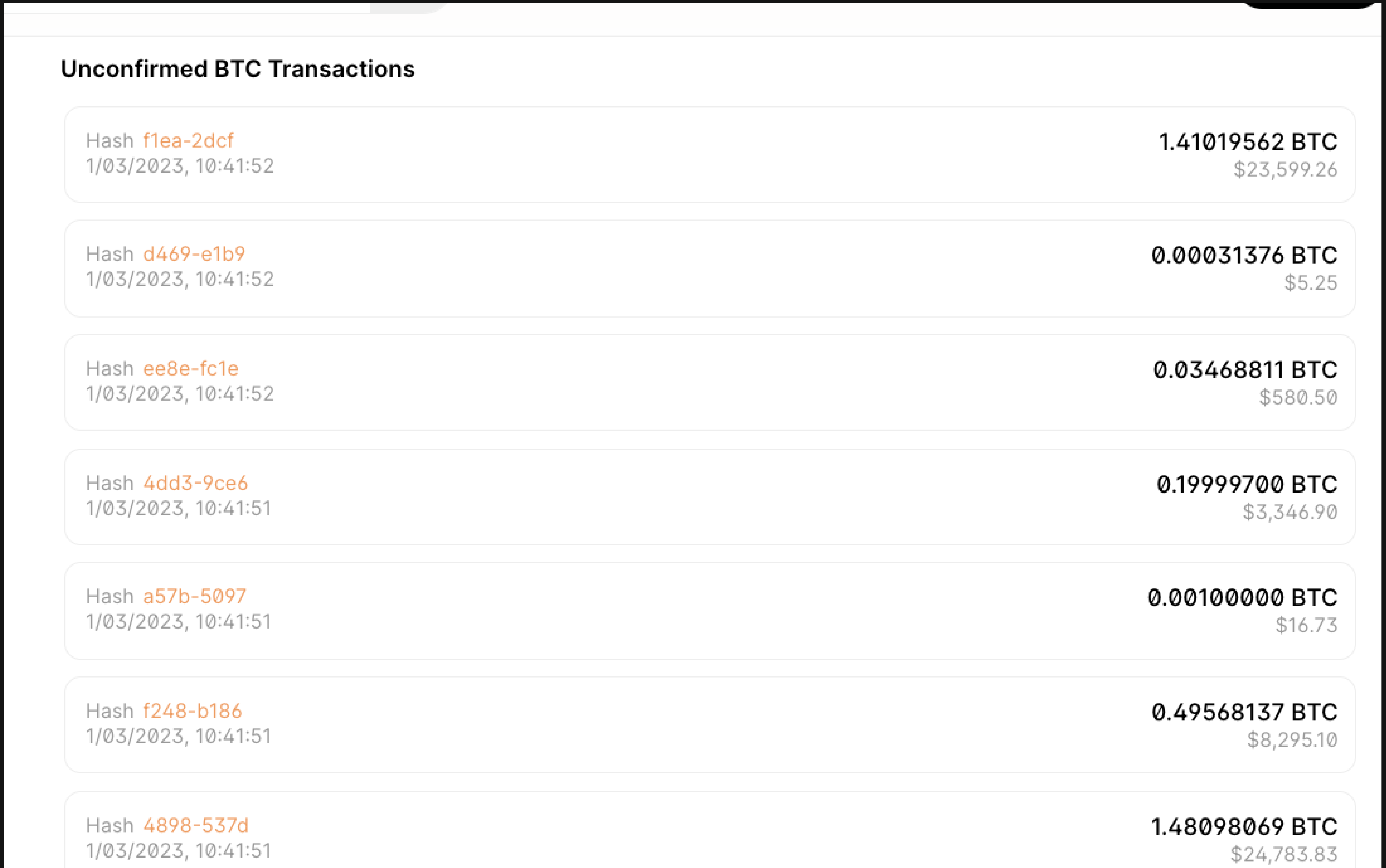
Peer to Peer Cash

Public Key Generation


Public Key to Bitcoin Address
Public Key
SHA256
RIPEMD160
"Double Hash" or HASH160
Public Key Hash (20 bytes/160 bits)
Base58Check Encode with 0x00 version prefix
Bitcoin Address (Base58Check Encoded Public Key Hash)

Peer to Peer Cash

Transactions are recorded on a public ledger visible to all parties without the need for a financial institution's involvement



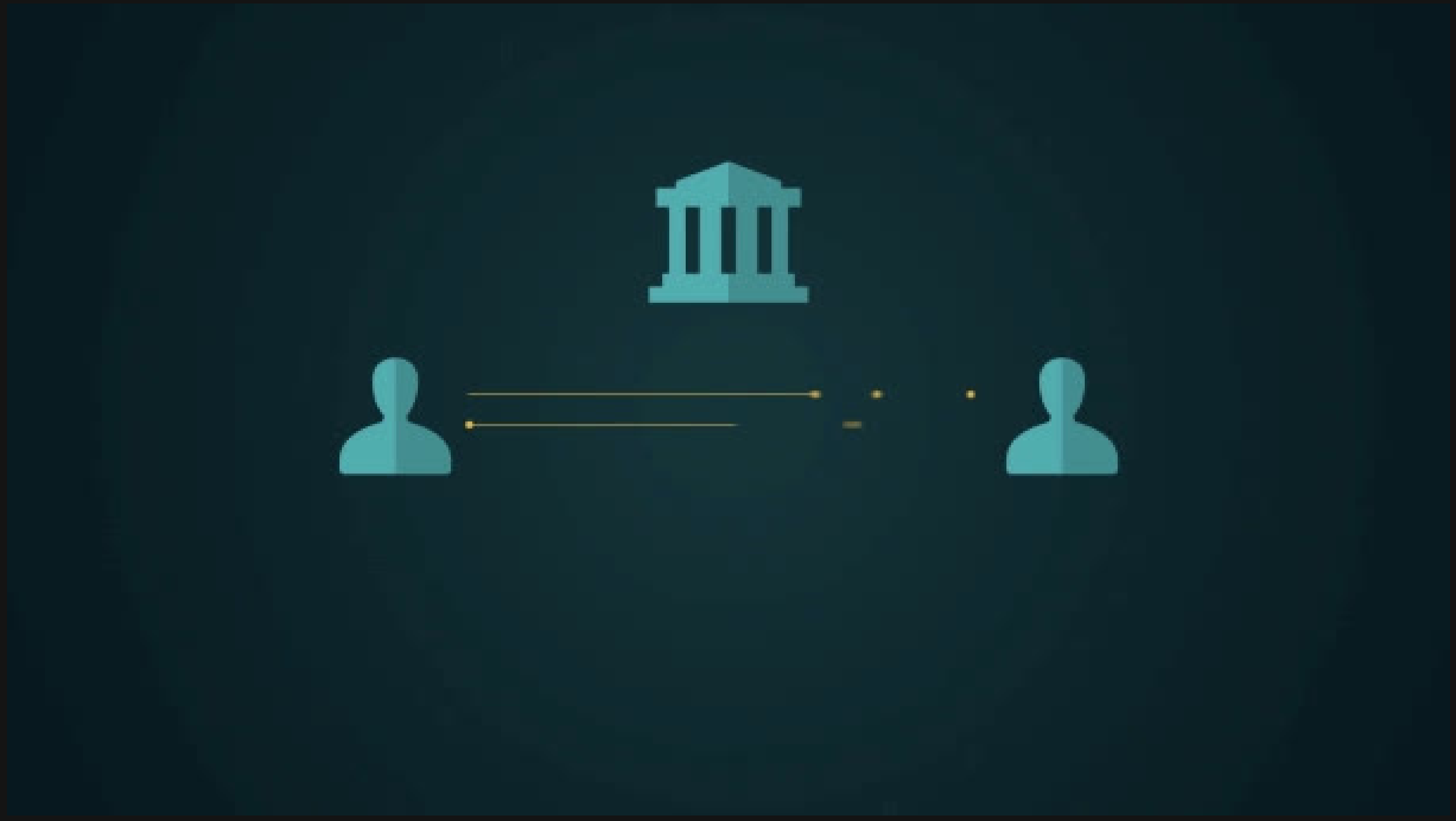
Unconfirmed BTC Transactions

Hash f1ea-2dcf 1/03/2023, 10:41:52	1.41019562 BTC $23,599.26
Hash d469-e1b9 1/03/2023, 10:41:52	0.00031376 BTC $5.25
Hash ee8e-fc1e 1/03/2023, 10:41:52	0.03468811 BTC $580.50
Hash 4dd3-9ce6 1/03/2023, 10:41:51	0.19999700 BTC $3,346.90
Hash a57b-5097 1/03/2023, 10:41:51	0.00100000 BTC $16.73
Hash f248-b186 1/03/2023, 10:41:51	0.49568137 BTC $8,295.10
Hash 4898-537d 1/03/2023, 10:41:51	1.48098069 BTC $24,783.83

Digital Signatures & Trusted Third Parties

Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending.

- Satoshi Nakamoto, Bitcoin Whitepaper

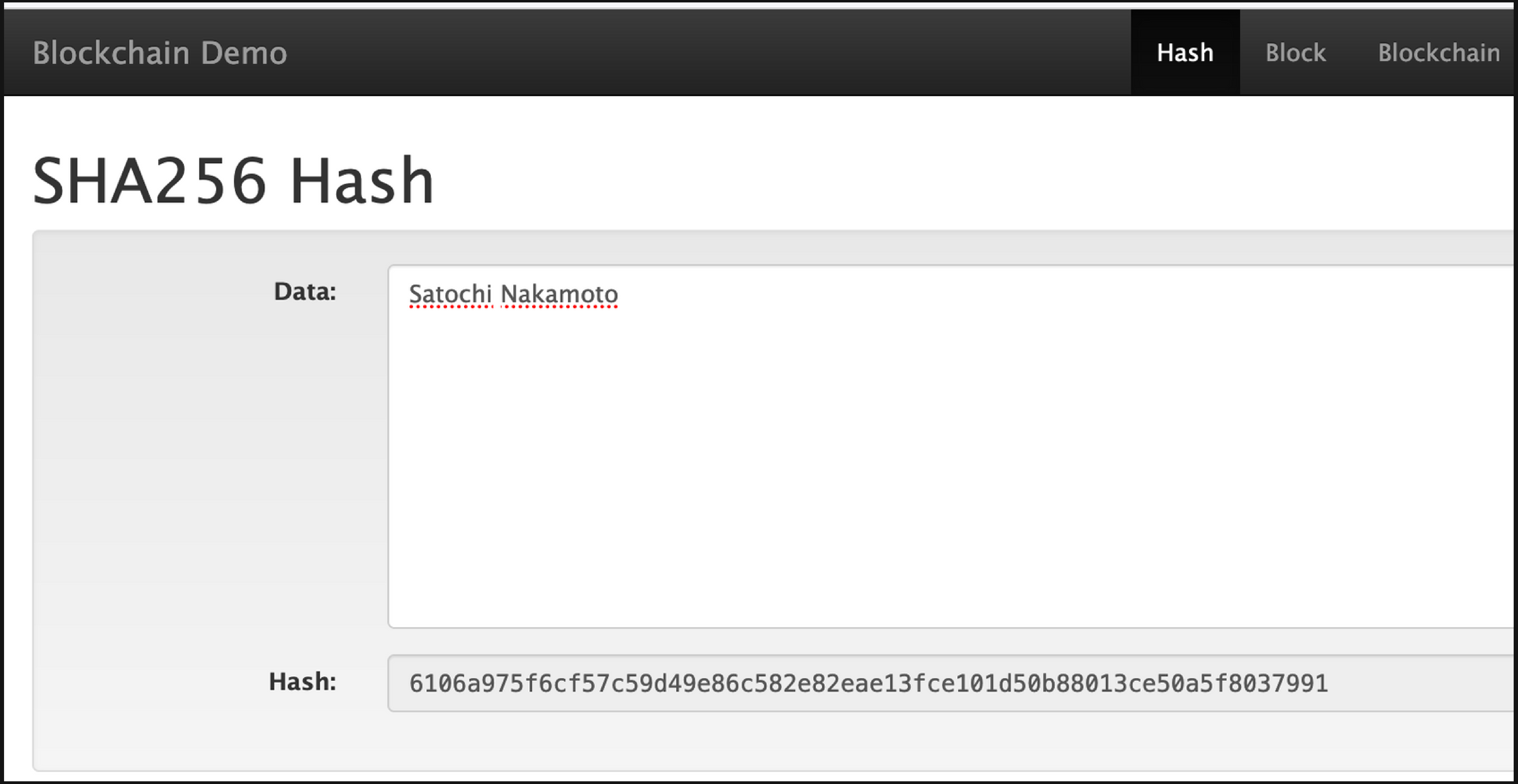


Digital Signatures & Trusted Third Parties

- Digital signatures are a means for the owner of a coin on the ledger to establish their intent to use that coin in a transaction.
- Digital signatures work by generating a unique hash of the message and signing it using the sender's private key
- The hash generated is unique to the message, and changing any part of the message will completely change the hash thus rendering the digital signature invalid.

Digital Signatures & Trusted Third Parties

Hash



SHA256 Hash

Data: Satochi Nakamoto

Hash: 6106a975f6cf57c59d49e86c582e82eae13fce101d50b88013ce50a5f8037991

Digital Signatures & Trusted Third Parties

- **A transaction consists of conditions under which a coin can be spent.**
    - Built using a flexible scripting language

- **Bitcoin uses the Elliptic Curve Digital Signature Algorithm (ECDSA) to incorporate keys into a transaction script**

- **Miners validate and record the transactions.**
    - **They are not privy to the contents**
    - They are paid a fee for their services
    - No trust needs to be given to them, hence, a simple third party

Peer to Peer Network

We propose a solution to the double-spending problem using a peer-to-peer network.

- Satoshi Nakamoto, Bitcoin Whitepaper

Peer to Peer Network

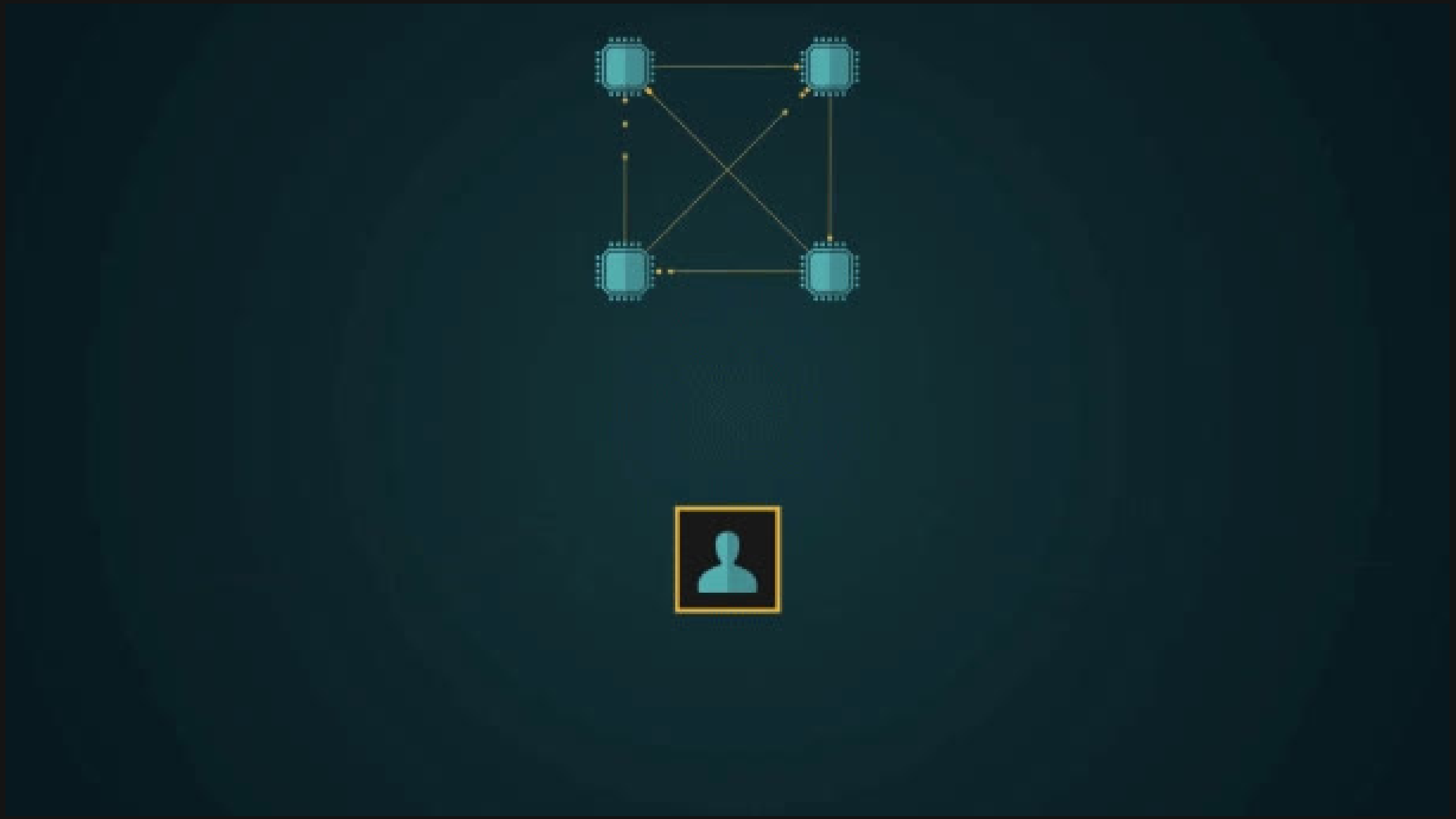
Double spending refers to the act of signing a coin to create a transaction and submitting that transaction to the network before using the same coin to create a different transaction which pays to a different recipient

- Peer-to-peer network of nodes gather, validate and timestamp all of the transactions that take place
- Accepts only the first-seen transaction that spends a UTXO.
- Each transaction can only be processed once and inputs used in a transaction are consumed

Timechain and Proof of Work

The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.

- Satoshi Nakamoto, Bitcoin Whitepaper

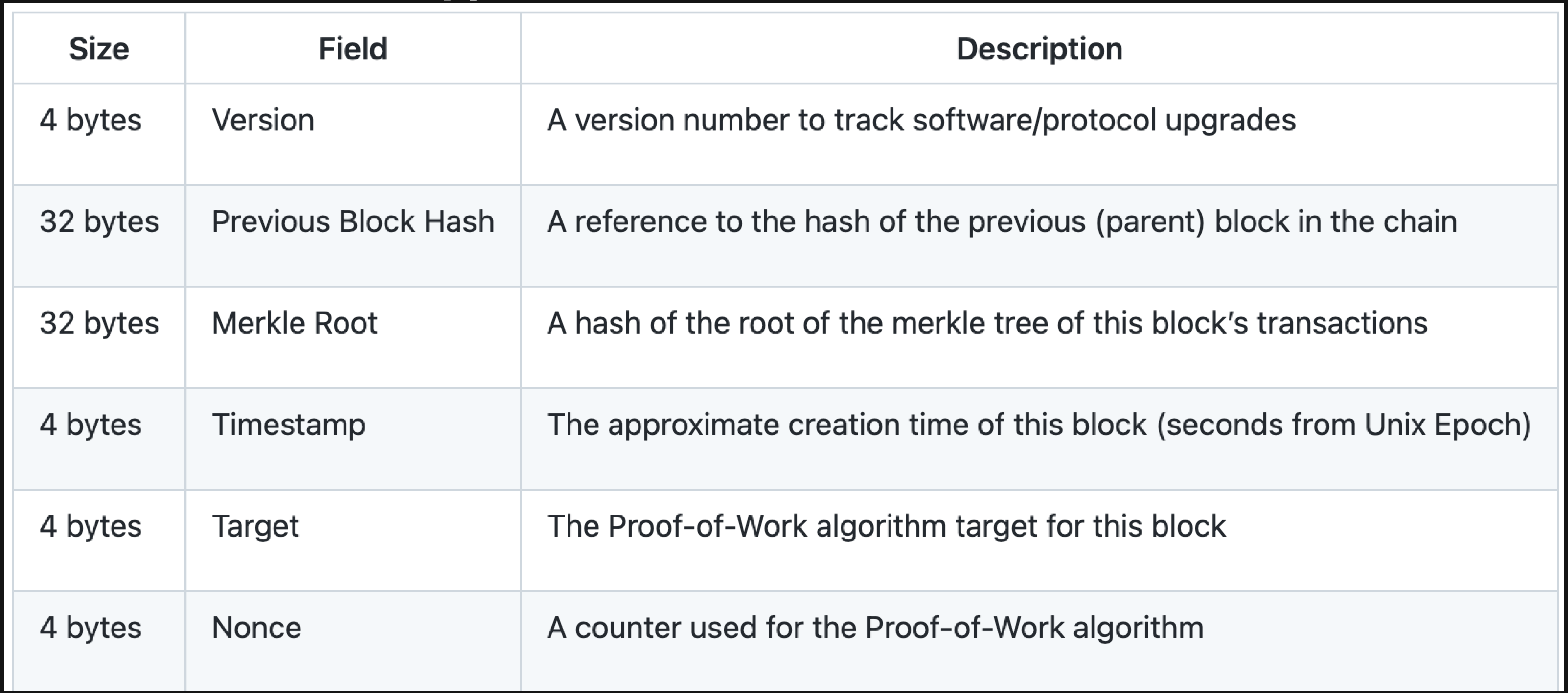


Timechain and Proof of Work

- As transactions are received into the network, nodes capture and collate them into logs
- These logs, or ‘blocks’, are made up of a timestamp applied to a sequential list of transactions
- represent a consensus agreement of the proof of both existence and validity of all the transactions they contain

Proof of Work

A typical Bitcoin block header

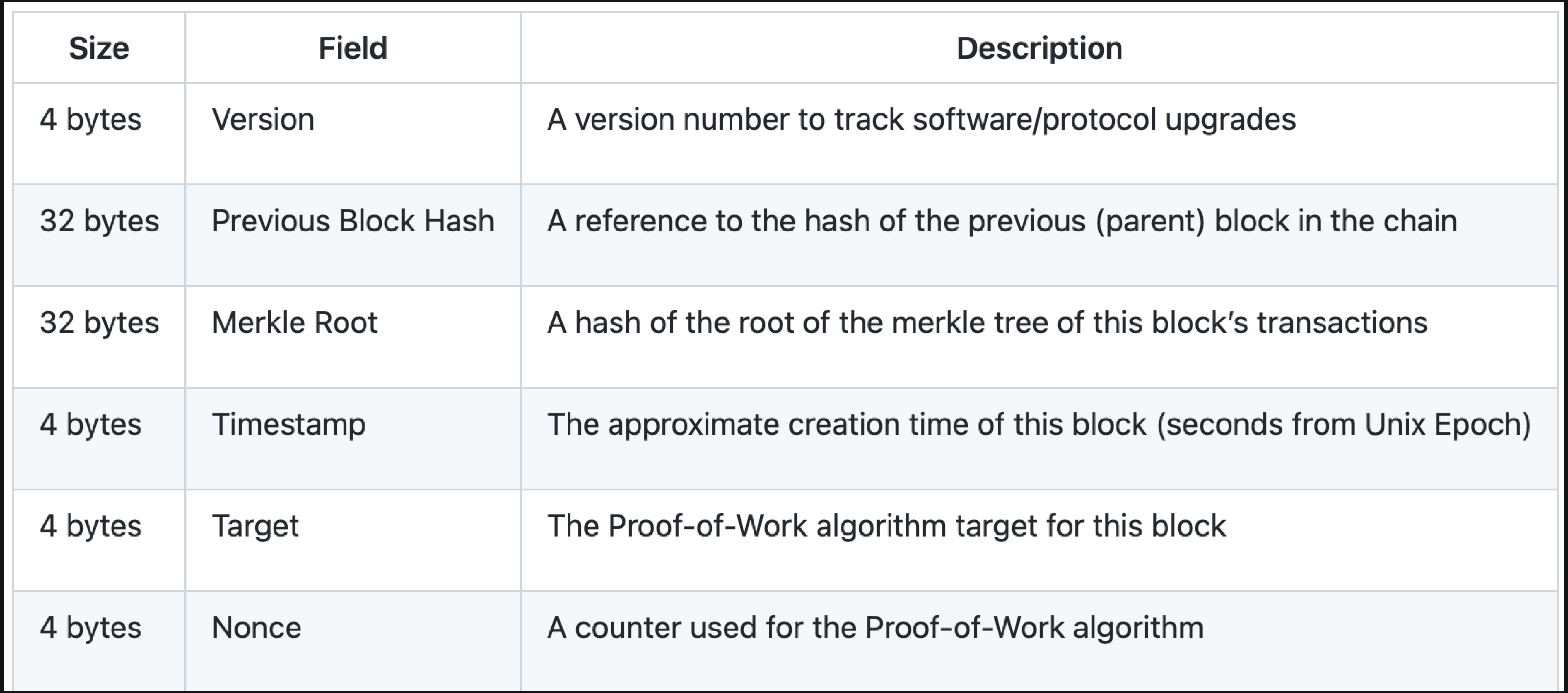


Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Target	The Proof-of-Work algorithm target for this block
4 bytes	Nonce	A counter used for the Proof-of-Work algorithm

- **All 6 fields must be filled with Nounce initialized to zero before minning begins**

Proof of Work

A typical Bitcoin block header



Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Target	The Proof-of-Work algorithm target for this block
4 bytes	Nonce	A counter used for the Proof-of-Work algorithm

- **All 6 fields must be filled with Nounce initialized to zero before minning begins**

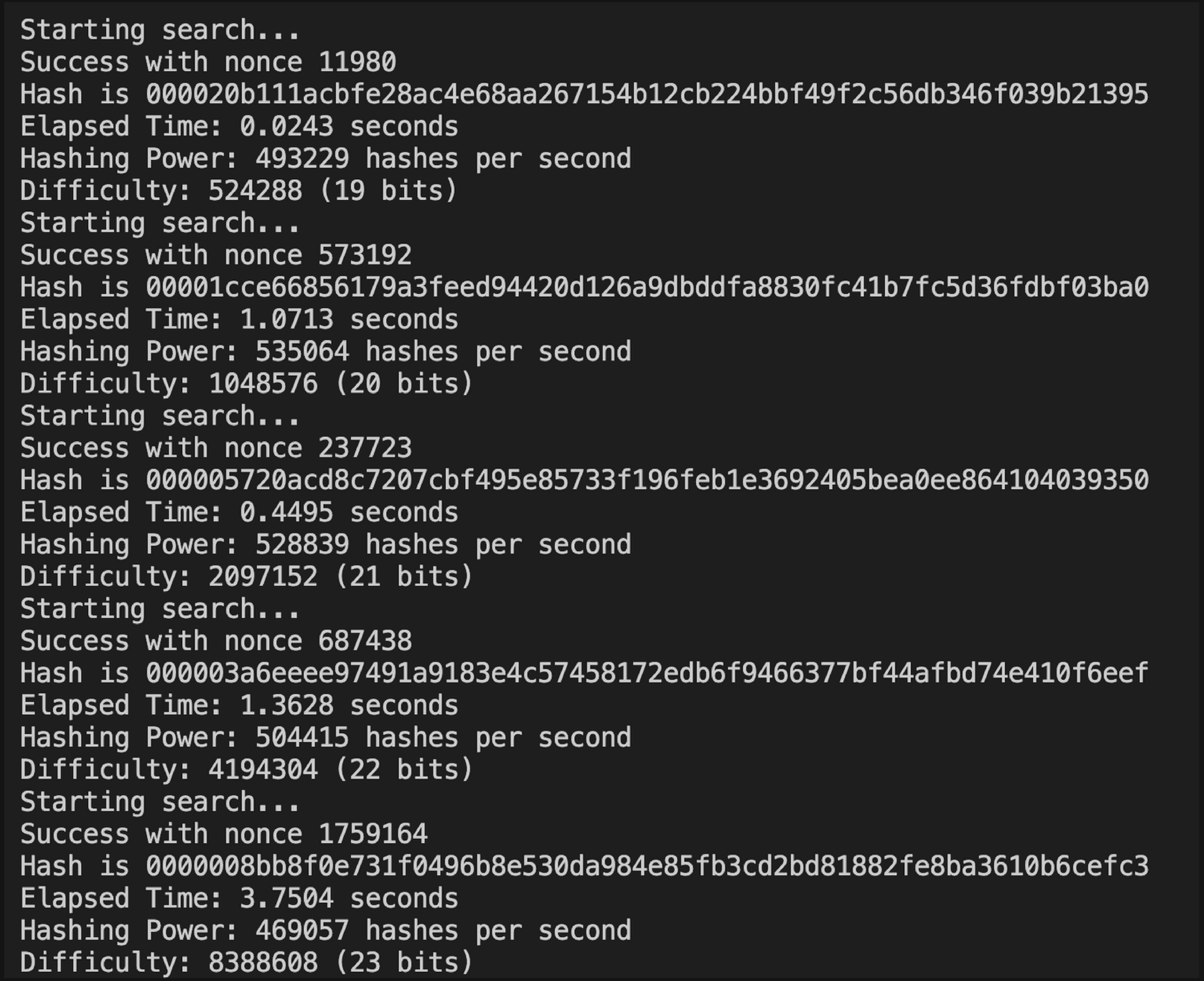
Proof of Work

- The miner calculates the hash of this block's header and sees if it is equal to or smaller than the current target.

- If the hash is greater than the target, the miner will modify the nonce (usually just incrementing it by one) and try again

- Miners have to try quadrillions of times before finding a nonce that results in a low enough block header hash

Simplified Proof of Work Algorithm



```
Starting search...
Success with nonce 11980
Hash is 000020b111acbfe28ac4e68aa267154b12cb224bbf49f2c56db346f039b21395
Elapsed Time: 0.0243 seconds
Hashing Power: 493229 hashes per second
Difficulty: 524288 (19 bits)
Starting search...
Success with nonce 573192
Hash is 00001cce66856179a3feed94420d126a9dbddfa8830fc41b7fc5d36fdbf03ba0
Elapsed Time: 1.0713 seconds
Hashing Power: 535064 hashes per second
Difficulty: 1048576 (20 bits)
Starting search...
Success with nonce 237723
Hash is 000005720acd8c7207cbf495e85733f196feb1e3692405bea0ee864104039350
Elapsed Time: 0.4495 seconds
Hashing Power: 528839 hashes per second
Difficulty: 2097152 (21 bits)
Starting search...
Success with nonce 687438
Hash is 000003a6eeee97491a9183e4c57458172edb6f9466377bf44afbd74e410f6eef
Elapsed Time: 1.3628 seconds
Hashing Power: 504415 hashes per second
Difficulty: 4194304 (22 bits)
Starting search...
Success with nonce 1759164
Hash is 0000008bb8f0e731f0496b8e530da984e85fb3cd2bd81882fe8ba3610b6cefc3
Elapsed Time: 3.7504 seconds
Hashing Power: 469057 hashes per second
Difficulty: 8388608 (23 bits)
```