Bitcoin: Design, Properties, & Challenges

Overview

- Brief History/Overview of previous attempts at Electronic Cash
- Primitives & Concepts relevant to Bitcoin
- Whitepaper Definition - “Purely peer-to-peer version of electronic cash”
- How well does the Whitepaper Definition hold in the real world?
- Attacks
- Protocol Developments & Scaling

An (Incomplete) History

- 1982, Chaum, Blind Signature for Untraceable Payments

  Blind Signatures from Banks as means of Payer/Payee Anonymity
- 1998, Wei Dai, B-money

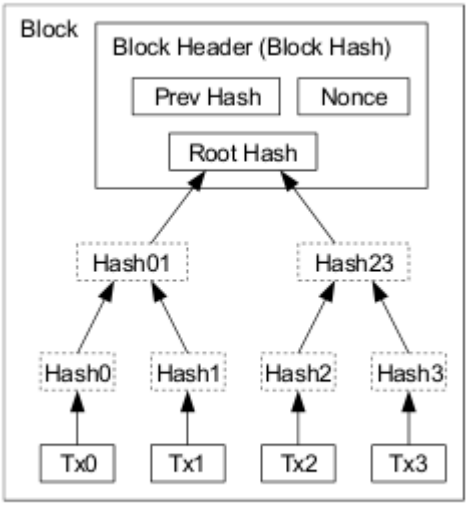
  Keypairs as Pseudonyms, Decentralized Issuance
- 2005, Ian Grigg, Triple Entry Accounting

  Records stored by payer/payee and in a central list

Primitives & Concepts

- ECDSA Keypairs (via secp256k1)
- Cryptographically secure hash functions (SHA256 & RIPEMD160)
- Proof-of-Work (Based on Adam Back's Hashcash)

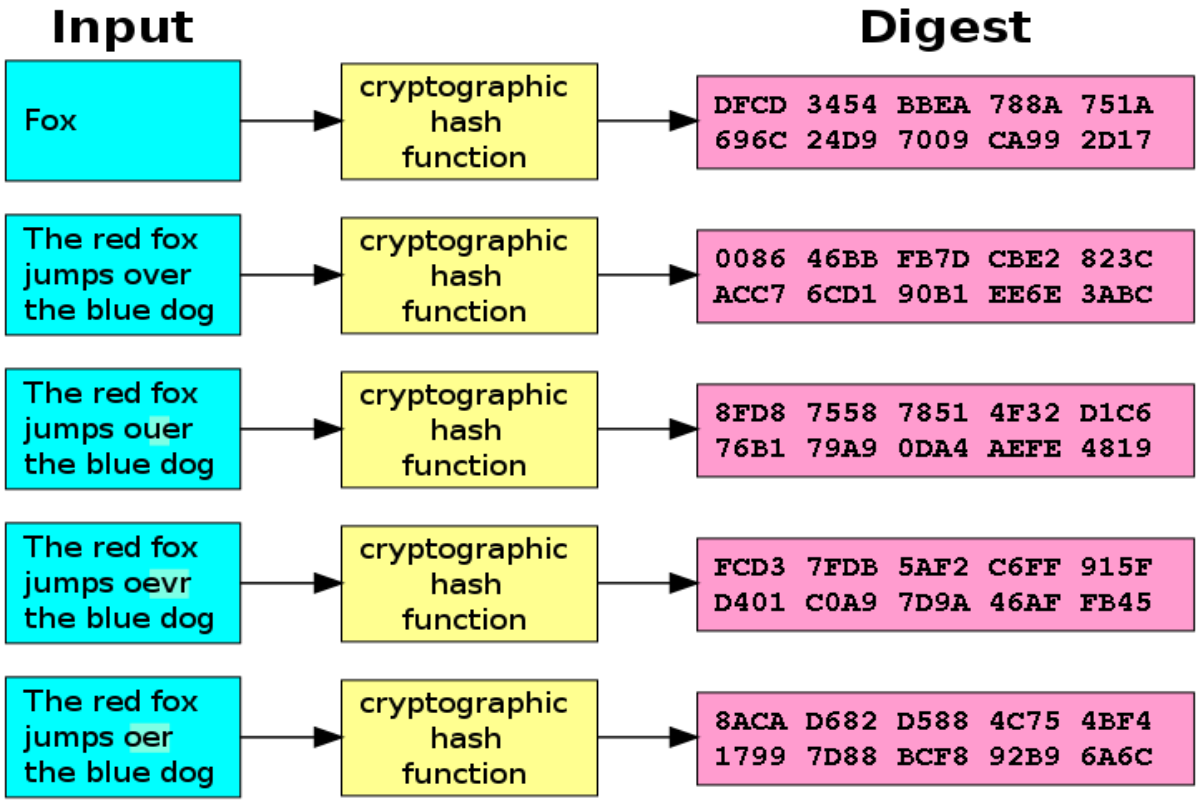
- “Blockchain”
- Merkle Trees
- Hash Chain
- Double spending



Transactions Hashed in a Merkle Tree

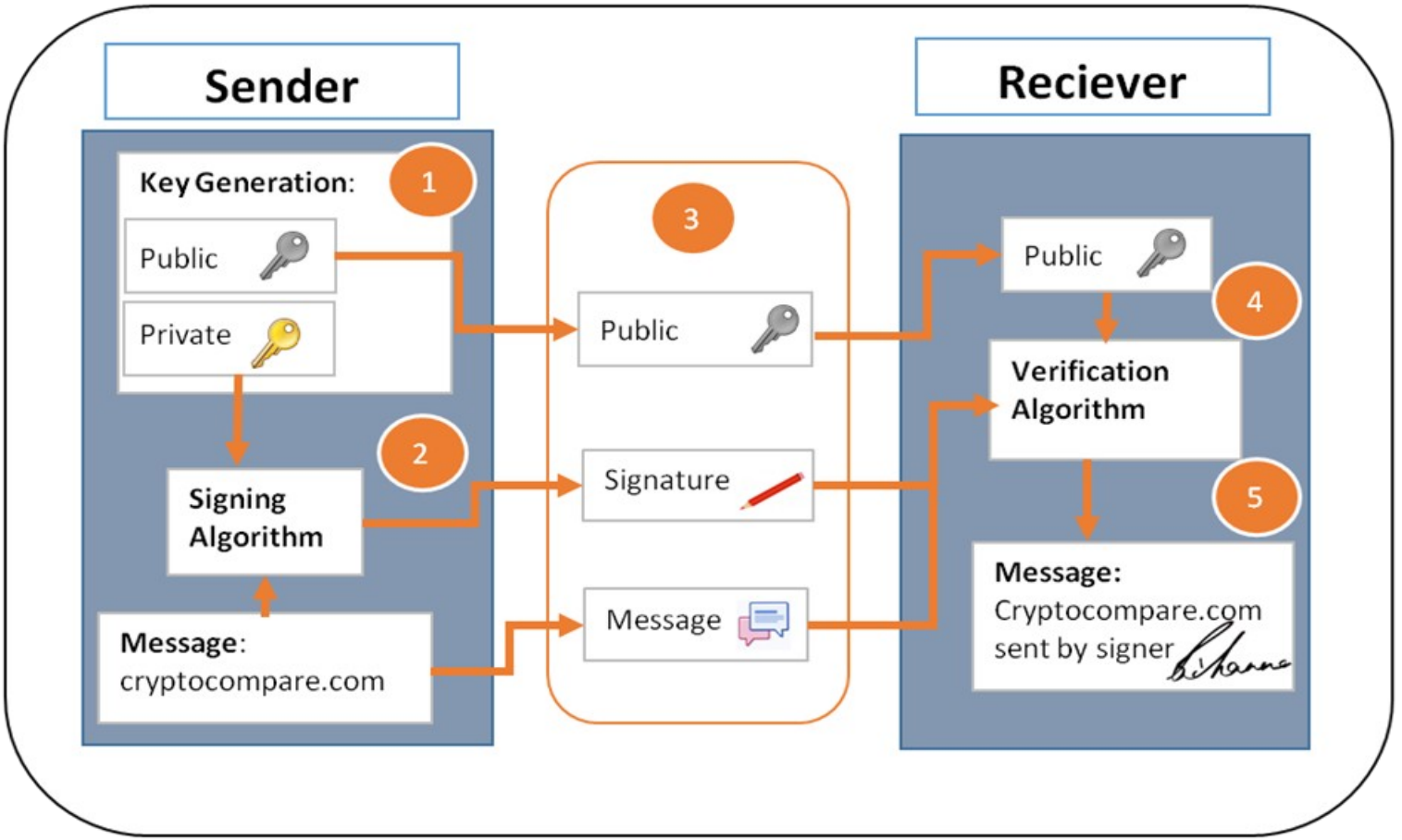
Primitives & Concepts

- Cryptographically secure hash functions (SHA256, RIPEMD160)
- Proof-of-Work (Based on Adam Back's Hashcash)

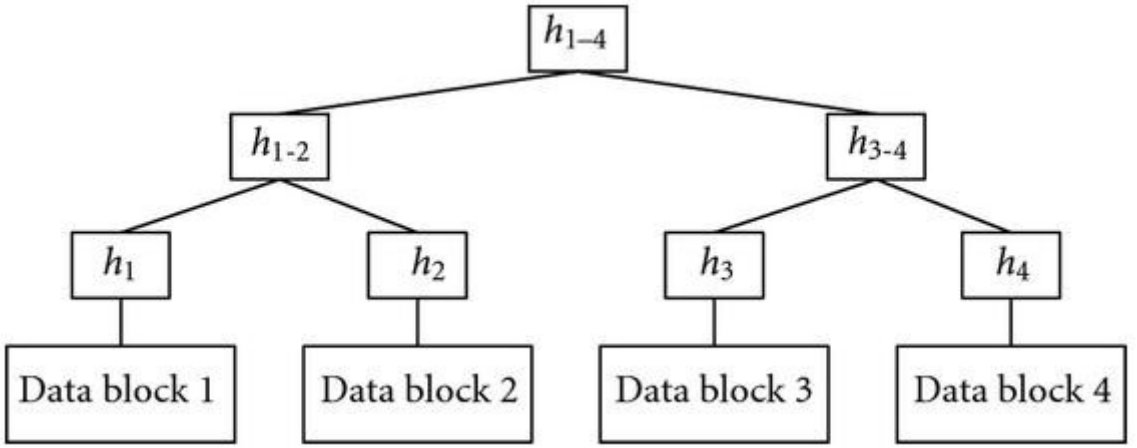


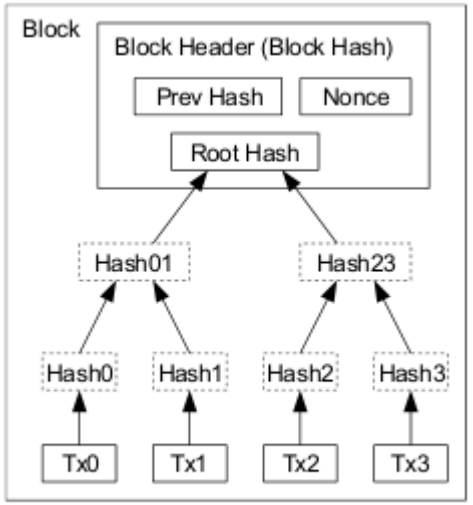
Primitives & Concepts

- ECDSA Keypairs (via secp256k1)

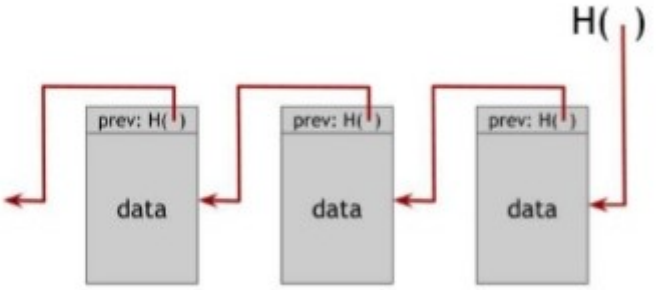


Primitives & Concepts





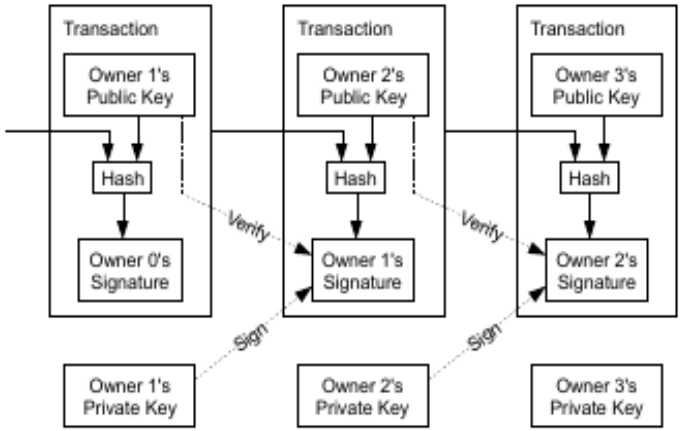
Transactions Hashed in a Merkle Tree



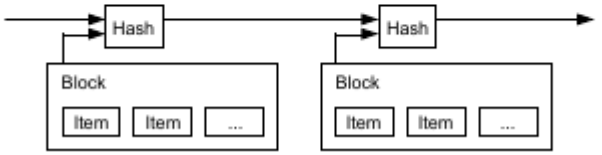
- “Blockchain”
- Merkle Trees
- Hash Chain
- Double spending

Satoshi Whitepaper

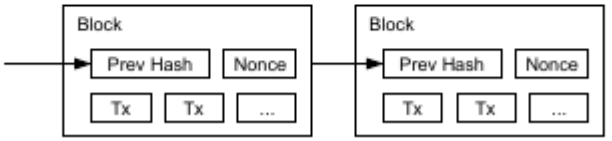
We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.



The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.



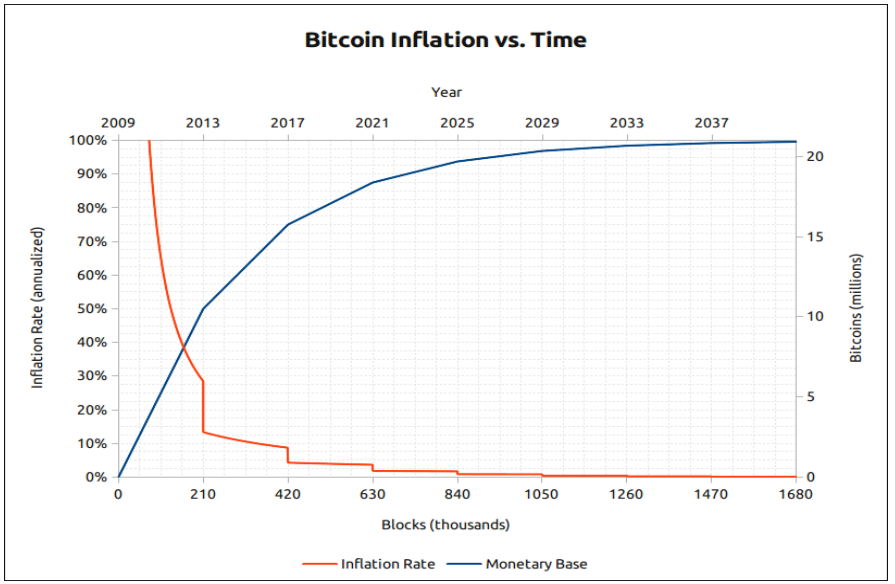
For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.

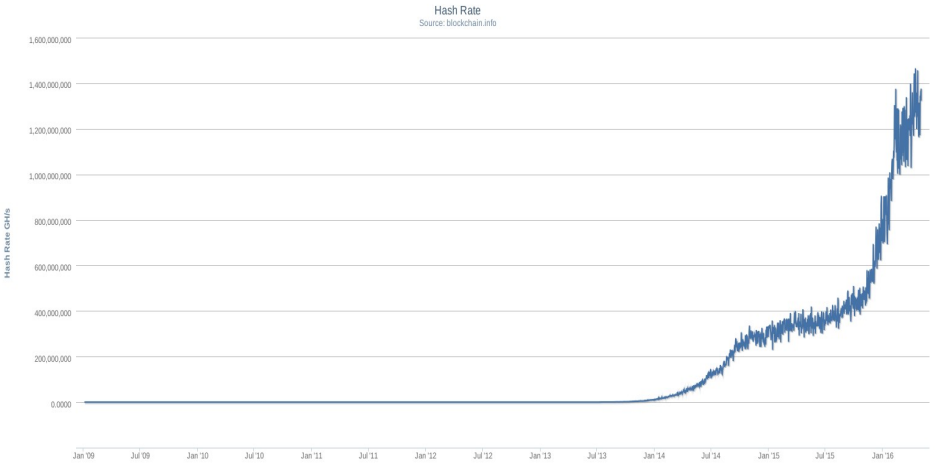


Important Network Parameters

- Difficulty Readjustment
- Block rewards
- UTXO set

- Every 2016 blocks (~2 weeks)
- Start at 50 Bitcoins, halved every 210,000 blocks (~4 years)





Real-World Network

- It works!
- Multi-billion dollar network
- Censorship-resistant asset & currency
- P2P network of 1,000s of full nodes
- 100s of MW of electricity dedicated to mining
- Development of ASICs
- Limited throughput
- **NOT ANONYMOUS**

Real-World Network II

Different levels of Participation:

- Solo Miner + Full Node
- Pooled Miner + Full Node
- Full Node
- Light/SPV Client
- Web Wallet
- Pooled Miner

Bitcoin as an industry:

- Exchanges
- Miners
- Standalone Wallets
- Web Wallets
- Blockchain Explorers
- Infrastructure

Real-World Network III

Complex Political Situation:

- ~5 people with commit access

- <10 entities control supermajority of hashrate

- Core-centric culture & development

- “Bitcoin is healthier than ever in that everything is contentious and hard” - Cory Fields

Attacks

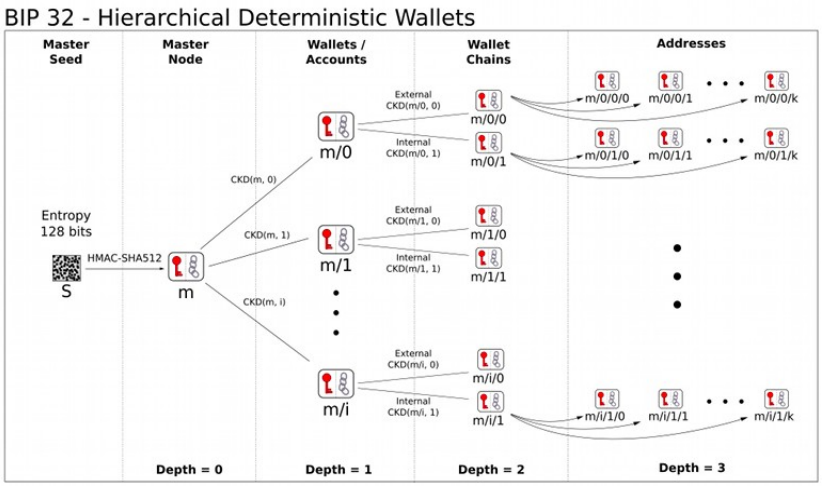
- 51% - Control a majority of the hashrate on the network
  - Censor transactions, double spend with decent probability, “Finney attack”
  - Worst-case scenario for network – Assumption made in whitepaper is “majority of nodes are honest”
- Block Withholding – Increase miner centralization
  - Reduce profit for miners not in your cartel
  - Similar strategy seen in Selfish Mining attacks

Attacks II

- Spam – Spend bitcoin to flood the network with many small, hard-to-process transactions
  - Now limits on maximum standard transaction size
  - In the past, miners & devs have communicated well to identify & filter spam (not ideal)
- Sybil – Control many nodes on the network
  - Can refuse to relay blocks and transactions, or relay only false blocks & transactions
  - DoS miners, pools, popular nodes, company nodes
  - Eclipse attacks, a variant, focus on monopolizing connections to a specific node

Protocol Developments

- Simplified Payment Verification (SPV)
  - Store and query only for information relevant to you
  - Popular for mobile devices
- Pay-to-Other Hashes!
  - Make use of Bitcoin scripting & opcodes for more complex transactions (via ScriptSig)
  - Multisignature transactions, timelocked outputs, etc.
- Hierarchical Deterministic Wallets



Scaling-Focused Developments

- Segregated Witness
  - Transaction Inputs/Outputs determine blockchain state; everything else is just needed for validation
  - Create a new Merkle Tree of validation data (scripts, signatures, etc.) and put the root in the block header
  - Downloading validation data is now optional (!)
- Invertible Bloom Lookup Tables (IBLT)
  - Most blocks contain transactions that full nodes are already aware of (set reconciliation); use IBLTs to compress transaction data, only decode new transactions, and then rebuild the block

Scaling-Focused Developments II

- Lightning Network
  - Implements Bidirectional Payment Channels; channel can be closed at any time
  - Lightning Network will allow for routing over such payment channels
- Pegged Sidechains
  - Non-Bitcoin blockchains that support atomic entry/exit from Bitcoin
  - Supporting work for more diverse & unified cryptocurrency ecosystem

References/Resources

- http://blog.koehntopp.de/uploads/Chaum.BlindSigForP
- http://www.weidai.com/bmoney.txt
- http://iang.org/papers/triple_entry.html
- https://bitcoin.org/bitcoin.pdf
- http://www.hashcash.org/
- https://bitcoincore.org/en/2015/12/23/capacity-increase
- https://github.com/bitcoin/bips/blob/master/bip-0141.me
- https://www.blockstream.com/sidechains.pdf

Thank You!

jharveyb@mit.edu

@jonhbit

bitcoin.mit.edu

bitcoin.org