NTNU

Norwegian University of Science and Technology

Cryptocurrencies

How Bitcoin Works

Colin Boyd

Department of Information Security
and Communications Technology, NTNU

Finse Winter School
May 2018

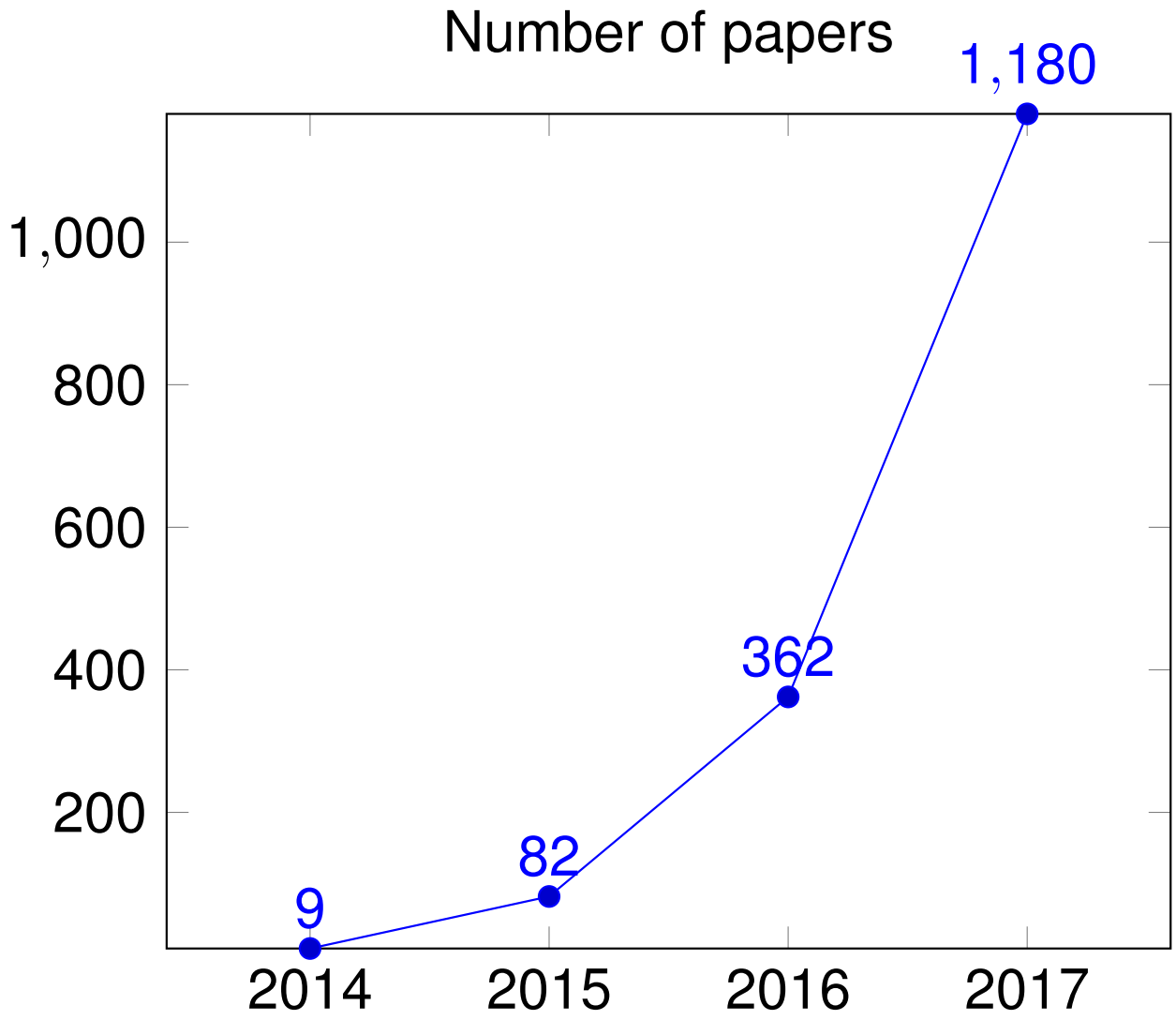
NTNU

The blockchain hype

> *Blockchain is causing a hype and optimism that has rarely been seen in the history of technology. It is celebrated as a new technological revolution, which will have at least as large an impact on society as the invention of the wheel, the steam engine or the Internet.*

Matthias Mettler, *Blockchain technology in healthcare: The revolution starts here*

The blockchain hype



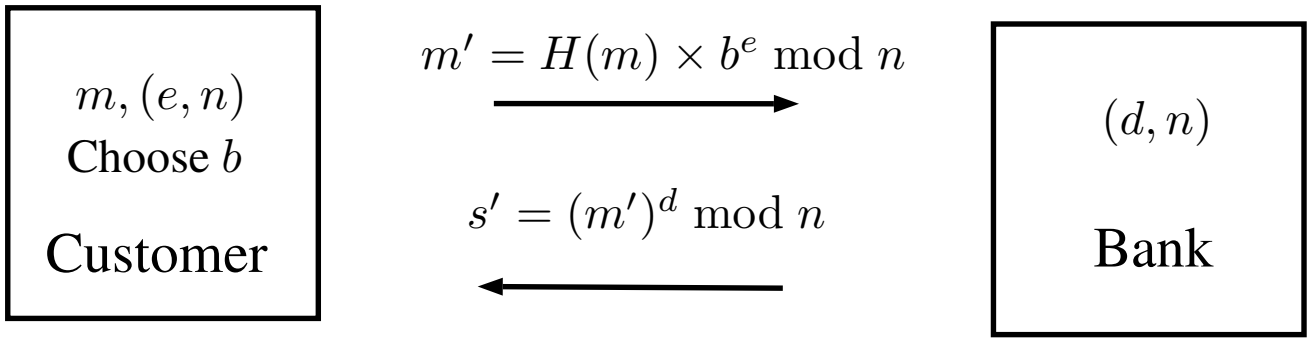
Outline

Digicash

- Company founded in 1989
- Electronic cash system developed by David Chaum
- Bankrupt in 1998

Blind Signatures

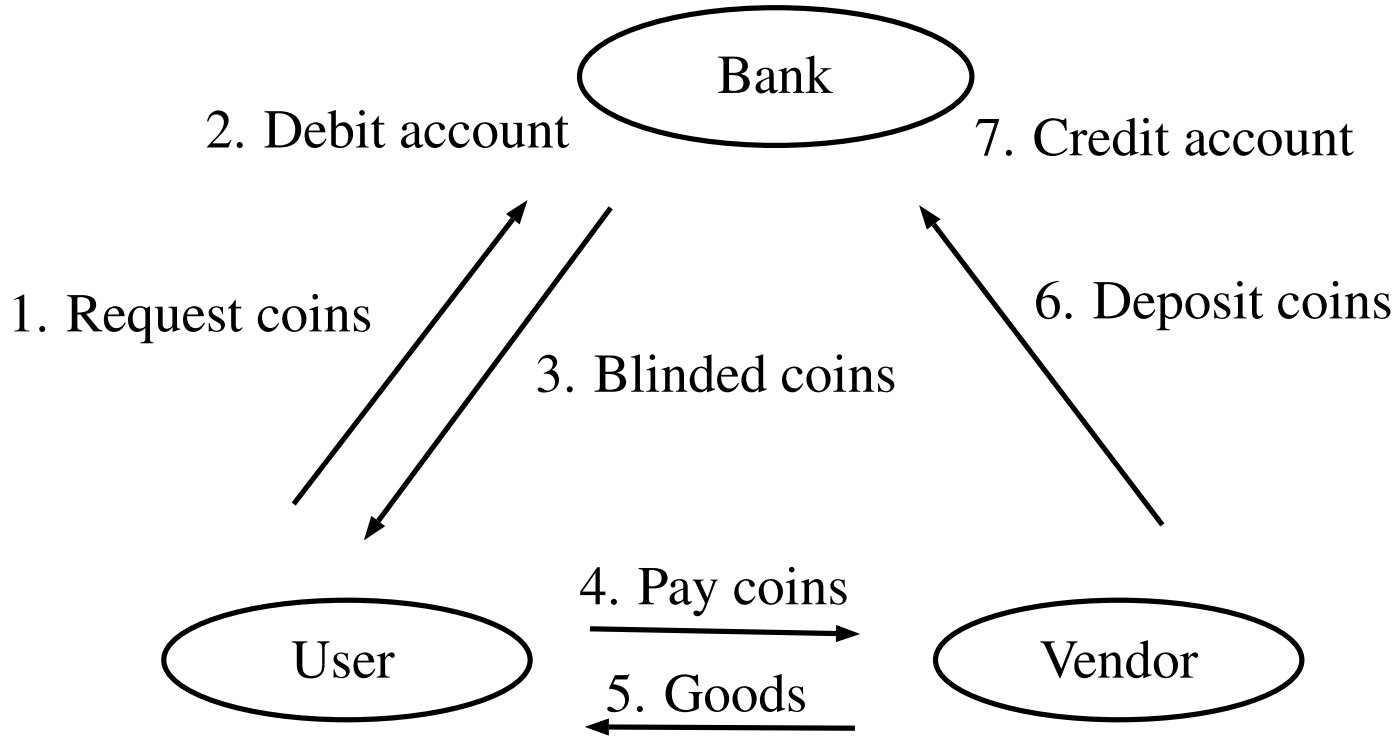
— Blind RSA signatures



$$s = s'/b \bmod n$$
$$= (H(m))^d \bmod n$$

— Also can be applied to ElGamal and similar signatures

1990s Ecash Protocol



— If user spends coin twice, bank can reveal identity (signature of fraud)

NTNU

Features of 1990s Ecash

- Anonymous (at least computationally)
- Only bank can issue coins
- Coins use local currency
- Double spending detection
- Failed commercially. Why?

Digital signatures

- A digital signature is a bit string which authenticates a message
  - Private signing key is used to generate each signature
  - Public verification key is used to verify each signature
- Main security property is *unforgeability* – signatures cannot be generated without signing key
- Bitcoin signatures use ECDSA with a specific curve – a modern efficient signature scheme

Digital signature algorithm (DSA)

— Standardised in FIPS 186-4

Parameters

— p, a prime modulus of L bits.

— q, a prime divisor of $p - 1$ of N bits.

— Valid combinations of L and N are: $(L = 1024, N = 160)$, $(L = 2048, N = 224)$, $(L = 2048, N = 256)$, $(L = 3072, N = 256)$.

— $g = h^{\frac{p-1}{q}} \bmod p$, where h is any integer, $1 < h < p - 1$.

— H, the SHA hash family variant which outputs an N-bit digest.

DSA algorithms

— Key generation
- Secret key x, random with $0 < x < q$;
- Public key $y = g^x \bmod p$.

— Signature generation
- Choose k at random with $0 < k < q$ and set

$$\begin{aligned} r &= (g^k \bmod p) \bmod q \\ s &= k^{-1}(H(m) - xr) \bmod q \end{aligned}$$

- The signature is the pair (r, s).

— Verification of signature (r, s) on m
- Calculate $w = s^{-1} \bmod q$. Set:

$$\begin{aligned} u_1 &= H(m)w \bmod q \\ u_2 &= rw \bmod q \end{aligned}$$

- Check $(g^{u_1} y^{-u_2} \bmod p) \bmod q = r$.

Randomness in DSA

— What are the unknowns in the signature element s?

$$s = k^{-1}(H(m) - xr) \bmod q$$

— What happens if the same k is used twice?

— Nobody told Sony about this in 2010

— Basis for double spending detection in 1990s cash

Exercise

Show that if the same k is used in two DSA signatures, then the private key x can be easily recovered from the two signatures and the messages they sign.

NTNU

Elliptic curves

- Elliptic curves are algebraic structures formed from cubic equations.
- An example is the set of all (x, y) pairs which satisfy the equation:

$$y^2 = x^3 + ax + b \bmod p$$

  This is a curve over the field $\mathbb{Z}_p$. Elliptic curves can be defined over any field.
- Once an identity element is added, a binary operation (like multiplication) can be defined on these points.
- With this operation the points forms a group over the elliptic curve, often called the *elliptic curve group*.

NTNU

ECDSA

- Elliptic curve variant of DSA (ECDSA) also exists in standard FIPS 186-4.
- Elliptic curve parameters are chosen from the NIST approved curves.
- Signature generation and verification is the same as in DSA except that:
  - the parameter q becomes the order of the elliptic curve group;
  - multiplication modulo p is replaced by the elliptic curve group operation;
  - after the operation on the group elements only the x-coordinate (an element in the underlying field) is kept.

ECDSA vs. DSA

- Because of the clever design of DSA, signatures using ECDSA are generally no shorter than signatures using DSA for the same security level.
- ECDSA signature size varies with the curve used. For approved curves this can vary between 326 bits and 1142 bits.
- ECDSA public keys are shorter than DSA public keys.

Exercise

Show that if $p + 1$ is divisible by 4 then $x^{p+1/4} \bmod p$ is a square root of x. Hence show how EC points over $\mathbb{Z}_{p^*}$ can be compressed to one element of $\mathbb{Z}_{p^*}$ plus one bit.

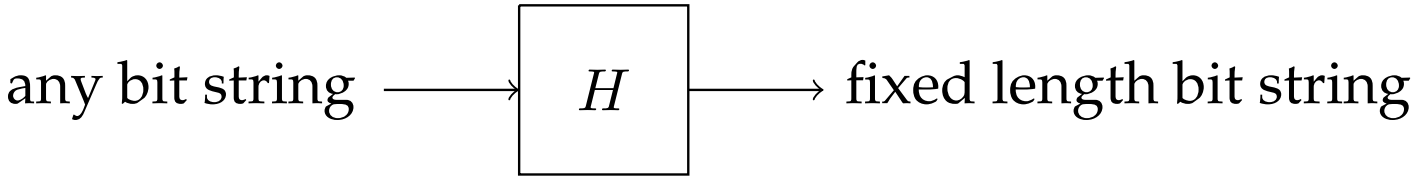
secp256k1

- Included in Standards for Efficient Cryptography published by Certicom Research
  `http://www.secg.org/sec2-v2.pdf`
- Points are solutions of

$$y^2 = x^3 + 7 \bmod p$$

  with $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$
- Curve order is prime, slightly smaller than 2^{256}
- Not included in the NIST curves standardised for ECDSA
- Public verification keys are two elements of 256 bits each, but point compression allows them to be 257 bits (or 33 bytes)

Hash functions



— Example SHA-256: output 256 bits (64 hex digits)

— SHA-256 hash of these slides[1]:

```
a60224e2bcd50cc84c8aebc11603d4d0
88c2356a93574e3f0ad46d323cef14cf
```

[1] Can this really be true?

Hash Collisions

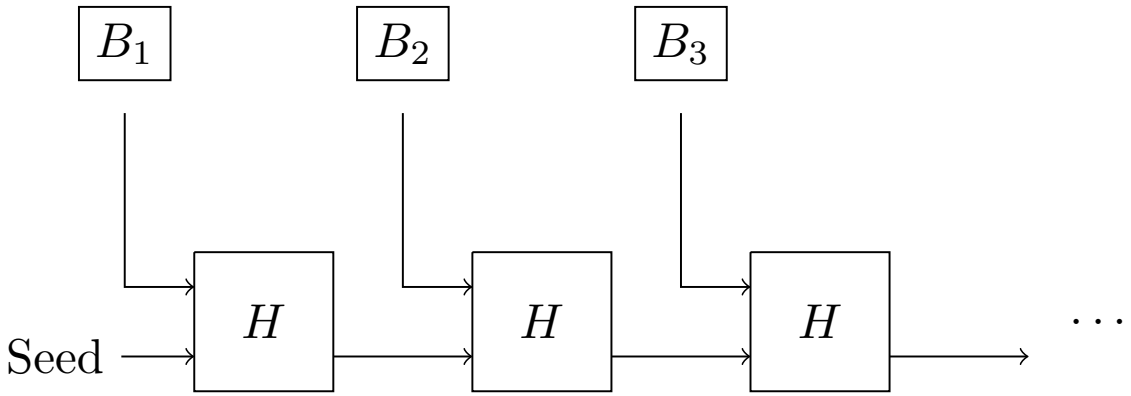
- $H(m_1) = H(m_2)$ but $m_1 \neq m_2$
- Collisions must *exist*

Fact

For a good hash function collisions are too hard to find

- We can authenticate m by authenticating $H(m)$

Hash chains



— Sequence of hashes. Each new hash input includes the previous hash.
— Cannot change (add, delete nodes) without finding a collision

Exercise

Given the end of a hash chain, V, show that changing any input value B_i without changing V results in hash collision

NTNU

Digital Timestamping using Hashchains

Hash chains used in cryptography for a long time

How to Time-Stamp a Digital Document*

Stuart Haber
stuart@bellcore.com

W. Scott Stornetta
stornetta@bellcore.com

Bellcore
445 South Street
Morristown, N.J. 07960-1910

Abstract

The prospect of a world in which all text, audio, picture, and video documents are in digital form on easily modifiable media raises the issue of how to certify when a document was created or last changed. The problem is to time-stamp the data, not the medium. We propose computationally practical procedures for digital time-stamping of such documents so that it is infeasible for a user either to back-date or to forward-date his document, even with the collusion of a time-stamping service. Our procedures maintain complete privacy of the documents themselves, and require no record-keeping by the time-stamping service.

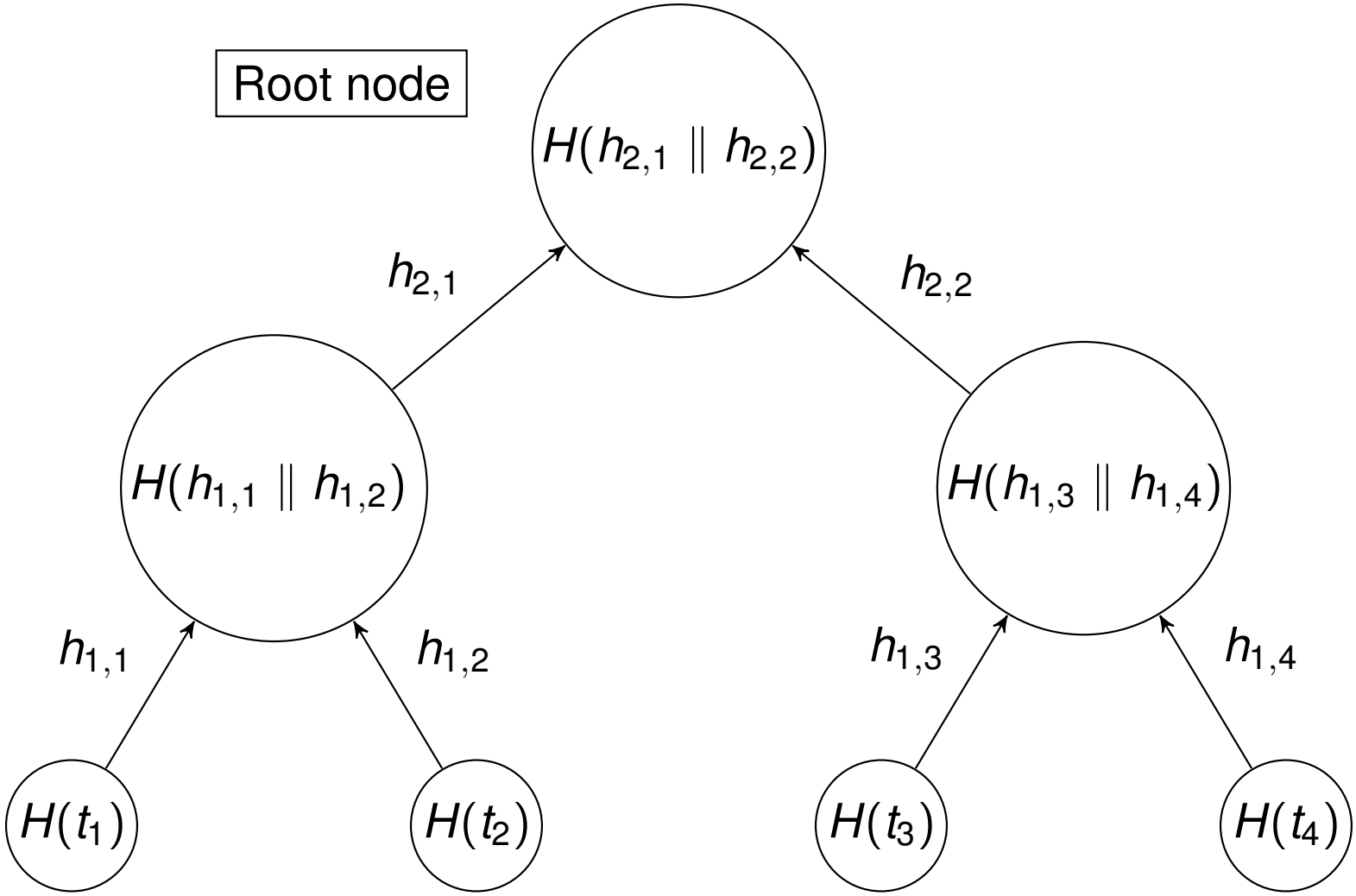
Published at Crypto 1990

NTNU

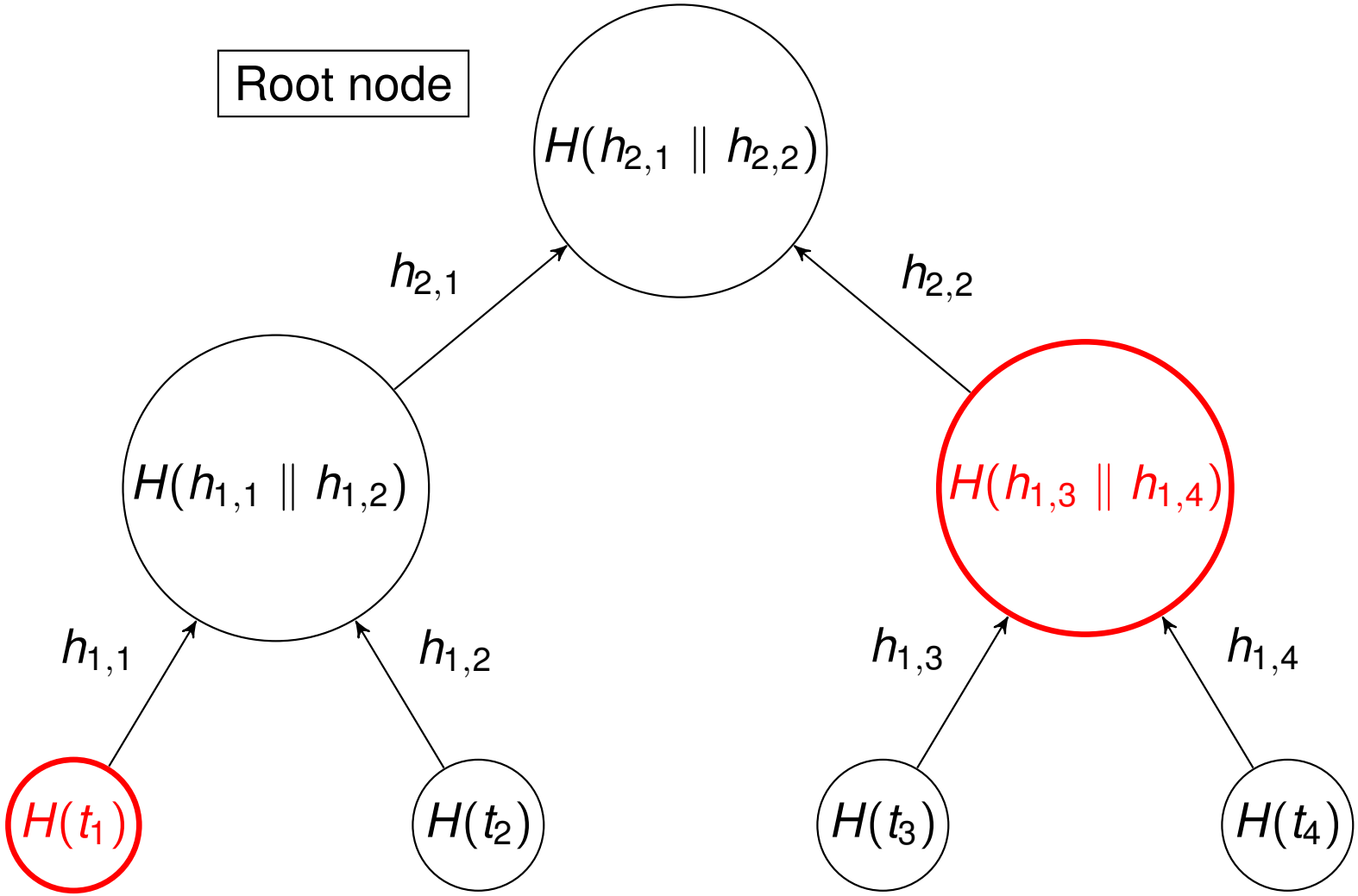
Merkle trees

- A generalisation of hashchain organised in a tree
- Authenticate by authenticating root of the tree
- Efficient proof of membership
- Efficient proof of non-membership by ordering the leaves

Merkle tree for 4 messages



Nodes required to check that t_2 is in tree



NTNU

Bitcoin origins

- Online proposal by Satoshi Okamoto late 2008
- First Bitcoin blocks formed 2009
- Protocol defined by implementation in software
- No central authority
- Not linked to any fiat currency

NTNU

Interfacing with the Bitcoin blockchain

- Block explorers
  - `https://blockexplorer.com`
  - `https://blockchain.info`
  - `https://www.blocktrail.com/BTC`
- Make a bitcoin node: install Bitcoin Core
- Toolkit: libbitcoin-explorer
  `https://github.com/libbitcoin/libbitcoin-explorer`
- Bitcoin testnet

Bitcoin addresses

- Bitcoin addresses are (hashed) public ECDSA verification keys
- Bitcoin payments go from one *bitcoin address* to another
- Addresses can be used once or multiple times
- Bitcoin uses multiple representations of bitstring, notably base58 and binary (hex)
- Bitcoin uses two different hash functions:
  - SHA256 with 256-bit output (used for ECDSA signing)
  - RIPEMD with 160-bit output (used in address checksum)

A typical Bitcoin address:

`1HnhWpkMHMjgt167kvgcPyurMmsCQ2WPgg`

Three versions of public key

— Binary version (in hex):

045901f6367ea950a5665335065342b952c5d5d60607b3cdc6c69a03df1a6b915
aa02eb5e07095a2548a98dcdd84d875c6a3e130bafadfd45e694a3474e71405a4

- 04 for uncompressed (03 is compressed)
- 2 x 32-byte coordinates of point on secp256k1

— Fingerprint:

b8268ce4d481413c4e848ff353cd16104291c45b

- Hash with SHA 256 and hash result with RIPEMD

— Bitcoin address:

1HnhWpkMHMjgt167kvgcPyurMmsCQ2WPgg

- add network version byte at front
- append 8-byte checksum
- encode in base58

Bitcoin transactions

- A normal transaction consists of one or more inputs and one or more outputs
- Each input has a value (number of bitcoins) and each output has a value

$$\sum \text{inputs} \geq \sum \text{outputs}$$

  Difference is the *transaction fee*
- Each input must spend all the value from some specified unspent previous input, known as an unspent transaction output, or UTXO

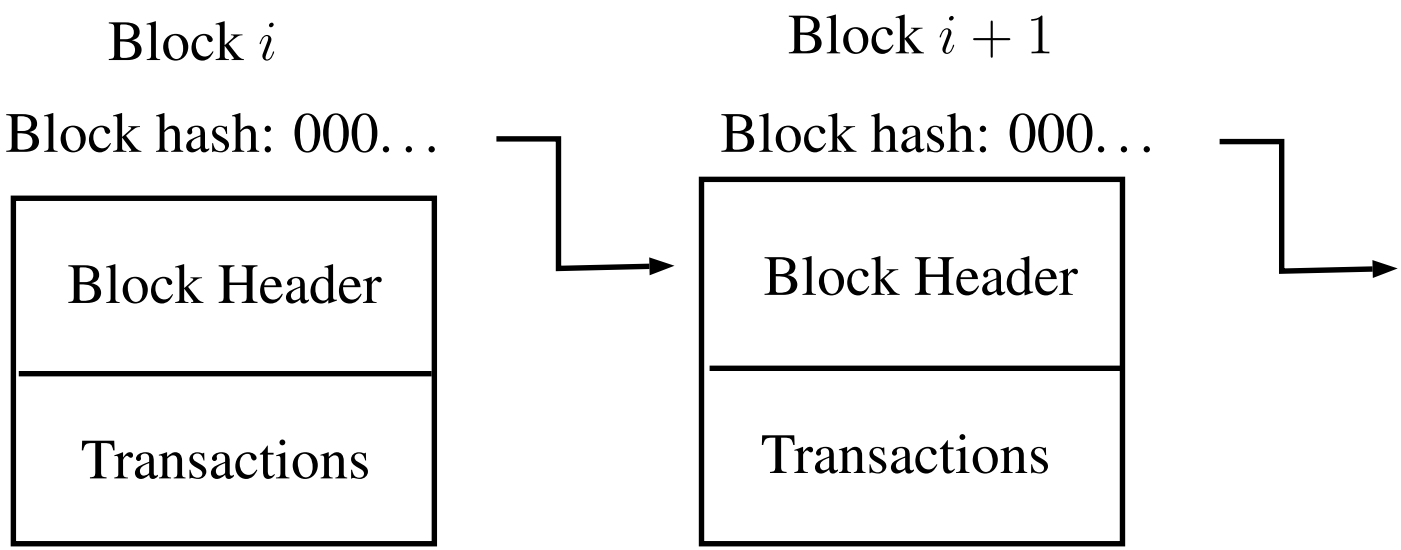
Bitcoin blocks

- A block consists of a header and a set of transactions
- Transactions are hashed into a Merkle tree
- The header (80 bytes) includes:
  - the double-SHA256 hash of the previous block header
  - the Merkle tree root of all the transactions in the block
  - a random nonce of 32 bits
- The combined size of each block cannot exceed 1 MB

Block header

Field	Size
Version	4
Previous Block Hash	32
Merkle Root	32
Block Time	4
Target (Difficulty)	4
Nonce	4

Chained blocks



NTNU

What is a Bitcoin?

- Bitcoins are units of value associated with a bitcoin address
- We can say that an address has x bitcoins if the total of the UTXO sent to that address is x
- The smallest amount of value allowed in a Bitcoin transaction is 10^{-8} bitcoins, known as one *Satoshi*.

NTNU

Coinbase transactions

- First transaction of every block
- No inputs
- Output is block reward + transaction fees for all transactions in block
- Block reward originally 50 Bitcoin, halves after every 210000 blocks (around 4 years) but goes to 0 after 21000000 Bitcoins has been issued (around year 2140)

Exercise

Show that after 32 halvings the block reward will be 1 Satoshi

NTNU

Puzzles

- Proposed in 1990s as “proofs of work”
- Dwork and Naor, Crypto 93 (see also Hashcash system of Back)
- First applications to provide a deterrent to spam email - must solve a puzzle to have your mail forwarded
- Also used in denial-of-service resistance: server generates a challenge and the client is required to solve a moderately hard puzzle based on this challenge.
- Should be easy to generate and easy to verify

Puzzles may be either *computation-bound* or *memory-bound*.

NTNU

Puzzle security properties

Difficulty: it should be moderately hard to solve a puzzle

Unforgeability: it should not be possible to generate valid puzzles without the required inputs

Non-parallelizability: it should not be possible to have multiple computers solve a puzzle in less time than a single computer could

Tuneable difficulty: can provide puzzles with different difficulty levels

Useful puzzles: the work done in solving a puzzle can be used for another purpose

Not all applications require all properties

Aura's puzzle for DoS mitigation

- Aura, Nikander and Leiwo, 2000.
- Server chooses nonce N_S and difficulty level Q. These are sent to the client.
- Client C generates nonce N_C. Needs to find X so that:

$$H(C, N_S, N_C, X) = \underbrace{00\ldots000}_{Q\ bits}\ Y$$

  Client C returns X together with nonce N_C.
- Puzzle verification uses only one hash call
- If H is a random function then client needs to make around 2^Q hash function calls before solving the puzzle

Mining

- A bitcoin block is a valid set of transactions
- A block is valid if it the hash of its header is small enough (has a lot of 0 bits at the start)
- A *miner* attempts to make a block valid – a computationally huge task (*proof of work*)
- Verifying a valid block is computationally very cheap - just a couple of hashes

Bitcoin consensus

Consensus is built by the community accepting that the longest valid chain is the correct blockchain

NTNU

Difficulty

- Smallest allowed difficulty, known as difficulty 1 is to find an input with 32 zeros at the start
- Mining difficulty re-adjusted every 2016 blocks to approximate 10 minutes per block
  (2016 x 10 minutes = 14 days)
- More accurate tuning is achieved by using a *target*, which the hash value must be below
- Mining evolution: CPU -> GPU -> FPGA -> ASIC
- Today all effective mining is done in *mining pools* – a huge industry

DIY Mining

Roll over image to zoom in

Bitmain

Bitmain Antminer S9 Bitcoin Miner, 0.098 J/GH Power Efficiency, 14TH/s

49 customer reviews | 52 answered questions

List Price: ~~$6,600.00~~
Price: $4,588.88 & **FREE Shipping**
You Save: $2,011.12 (30%)

In Stock.
This item ships to **Norway**.
Ships from and sold by Merkamerica Inc.

- Bitcoin Mining Hash Rate: 14.0TH/s ±5%
- Power Consumption: 1372W ±10% (Power supply not included)
- Most Power Efficient Bitcoin Miner: 0.098 J/GH ±7%
- Built-in web management portal - No separate host computer or software required
- Power supply sold separately - AntMiner APW3++

Share

Buy new: $4,588.88

Qty: 1

$4,588.88 + Free Shipping
In Stock. Sold by **Merkamerica Inc**

Add a Protection Plan:

4-Year Protection for $39.33
3-Year Protection for $39.26

Add to Cart

Turn on 1-Click ordering for this browser

Ship to:

Norway

Industrial Scale Mining

BITCOIN MAGAZINE NEWS GUIDES PRICE & DATA OPINION TECHNICAL ARC

MINING

Home › Articles › Bitmain Reveals Plans for Major Bitcoin Mining Data-Center in Northwestern China

by Aaron van Wirdum
Staff Writer
Nov 10, 2016 11:57 AM EST

Pending on po.et

What is Po.et?

Tweet

Bitmain Reveals Plans for Major Bitcoin Mining Data-Center in Northwestern China

NTNU

How Much Electricity does Mining Use?

— Estimates vary a lot, from the size of a small country to a medium size power plant.

— Top down approach: how much is it worth spending? (See next slide)

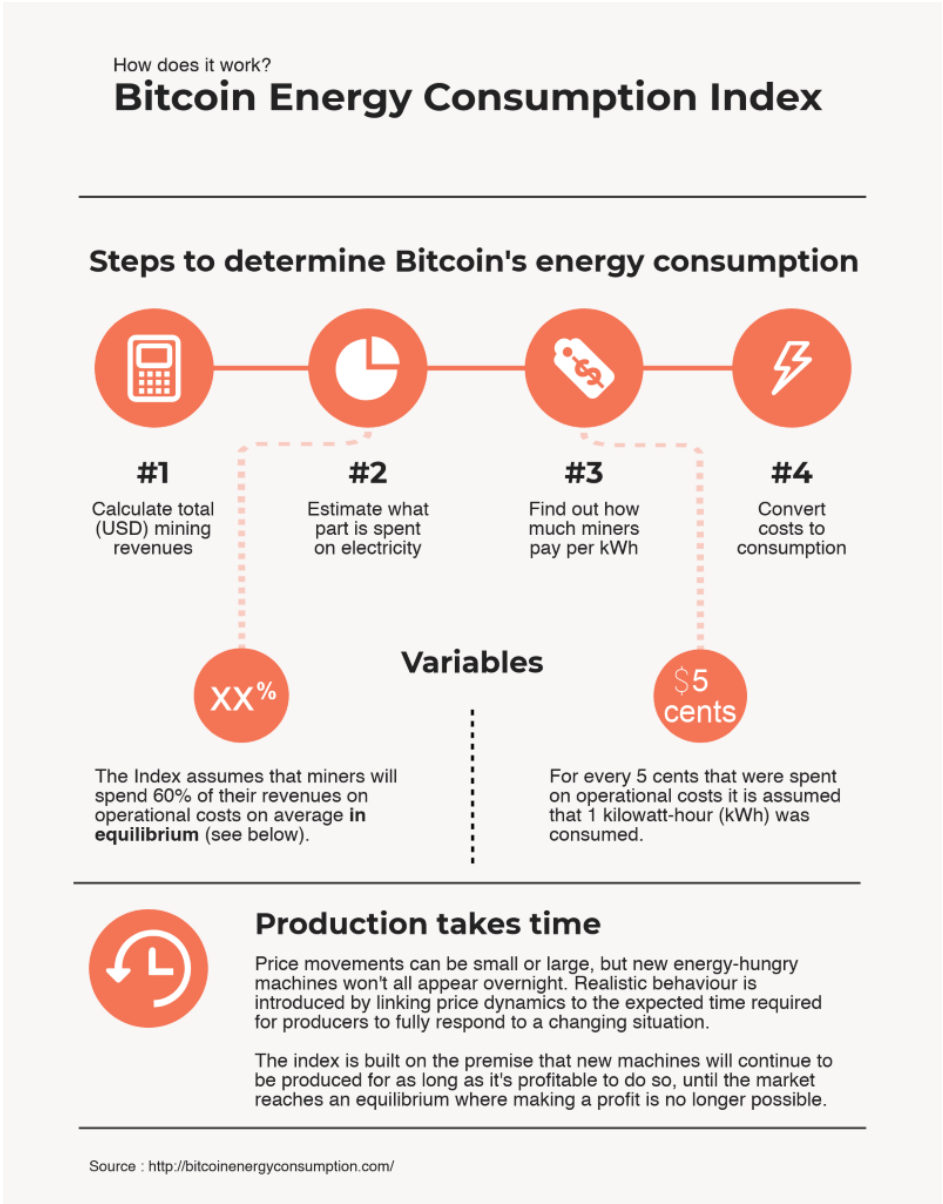
— Bottom up: how much energy does mining hardware need per hash?

— Example:

- Total Bitcoin hash rate: $\approx$ 30 million TH/s
- AntMiner S9 hash rate: 14 TH/s
- Power for the AntMiner S9: 1375 W

Gives $30/14 \times 10^6 \times 1375W \approx 3\text{GW}$

How Much Electricity does Mining Use?



https://digiconomist.net/bitcoin-energy-consumption

NTNU

Indelible Digital Graffiti

- The Bitcoin blockchain contains many messages hidden in bogus (unspendable) Bitcoin addresses:
  - news items
  - original Bitcoin paper of Satoshi Nakomoto
  - advertising
  - political and religious messages
  - portrait of Nelson Mandela
  - ...
- Can also be used as a notary service
- Available as a service for around $1 per kB

See `https://proofofexistence.com` and `http://www.cryptograffiti.info`

Storing data on the blockchain

- Choose some fields non-randomly;
  - Public key bits
  - Coinbase parameter
  - In payment script using `RETURN` operation
- `RETURN` operation is now approved method – allows pruning of data from blockchain
- Example: there is some data on the Bitcoin testnet in this transaction:

  `ccc1a6bd109e410fa7a23552c637c42c0a5b83c8fdd7cbdf7c8da00b2f4c9b87`
- Can also be used for “proof of burn” to prove that coins are destroyed

Question

Is it reasonable to allow anybody to put any data into the public and immutable blockchain?

NTNU

Forks

- A fork occurs when two valid blocks are formed which extend the blockchain.
- *Temporary forks* happen when two miners solve two valid blocks extending the chain at almost the same time. Leads to stale or orphaned blocks
- Example: Orphaned block

  `8a91366c2da8ce175a0bd477f330240ba67526da7e6452dd10de10bdbf95b0cb`
- *Soft forks* change the rules for valid blocks to be stricter. If majority follow new rules then chain will remain intact.
- *Hard forks* change the rules for valid blocks to be more lenient. Leads to a permanent split in the chain

NTNU

Bitcoin Cash - A Hard Bitcoin Fork

- Dispute about how best to handle more capacity in Bitcoin blockchain
- Fork in Bitcoin and Bitcoin Cash at block 478558
- Bitcoin Cash allows blocks up to 8MB
- Bitcoin Cash difficulty reduced at block 478577 and next 5 blocks

NTNU

What are security threats?

- Theft of coins (value)
- Double spending
- Integrity of coins (value)
- Loss of availability
- Privacy violation

Double spending

- Why not transfer the same value twice?
- A transaction will not be valid if the address does not have sufficient value
- To be sure that a transaction is valid it must be on the blockchain and stay on the blockchain
- Usually recommended to wait until 5 more blocks are added after the one with the transaction

NTNU

Scaling

- Total Bitcoin blockchain size is around 165 GB
- Pruning of spent UTXOs massively reduces storage
- Many blocks are almost at capacity
- Maximum transaction rate is around 5-6 transactions per second

Exercise

What is the maximum rate at which the blockchain size can increase per year?

NTNU

51% Attacks

- What if one party controls the majority of the hash power?
- Make history
- What is consequence of a 51% attack?

NTNU

Selfish Mining (33% Attacks)

- Keep mining in secret
- Rewrite history
- What attacks are possible?

NTNU

Formal modelling

Garay, Kiayias and Leonardos (Eurocrypt 2015 and later) prove formal properties with the assumptions that the adversary does not control too much of the hashing power.

Persistence: all nodes agree on confirmed transactions

Liveness: transactions will be confirmed

Led to design of Cordano (Ourorobos)

NTNU

Bitcoin Scripts

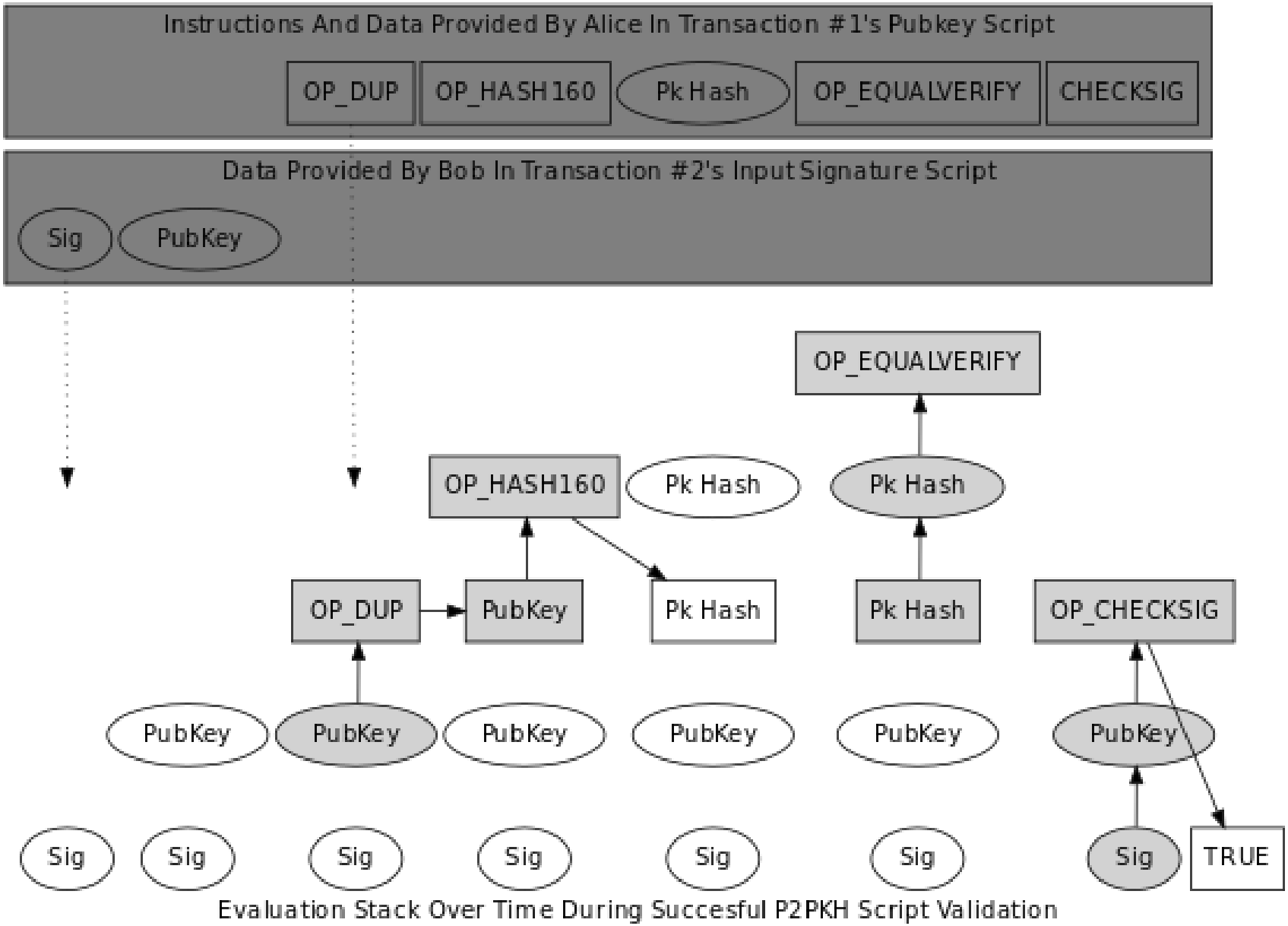
- Bitcoin uses a simple stack-based language to validate transactions
- Data items are read in and put on top of the stack
- Operations take arguments from the top of the stack
- Limited operations with no loops
- Operations include arithmetic, basic logic, hashing and signature verification
- Operation `RETURN` allows any data, of length 40 bytes, to be recorded in a transaction

Pay to Public Key Hash (P2PKH) Scripts

- Most Bitcoin transactions use this basic script
- The output script specifies what public key needs to sign in order to obtain the funds.
- When redeemed the correct public key and signature are provided

```
<Sig> <PubKey> OP_DUP OP_HASH160 <PubkeyHash>
OP_EQUALVERIFY OP_CHECKSIG
```

P2PKH Stack Evolution (Source: `bitcoin.org`)



Bitcoin Explorer – `libbitcoin`

- Command line tool for Bitcoin (and testnet)
- Runs on Linux, Windows, OSX (executables available)
- Generate keys, addresses and transactions
- Interface with online Bitcoin blockchain
- Demo:
  - Use configuration file to specify Bitcoin testnet
  - Obtain some testnet coins, for example: `http://bitcoinfaucet.uo1.net/`
  - Make a payment from A to B with a P2PKH script

Pay to Script Hash (P2SH) Scripts

- Allows transactions to go to a script address
- When redeemed, a script must be supplied which maps to the hash in the payment output
- The script is run on the other inputs and must return `TRUE`

```
OP_HASH160 <Hash160(redeemScript)> OP_EQUAL
```

Multisignatures

- Probably should be called multiple signatures — each signature is added separately
- Most common P2SH script
- Example of 2 out of 3 signing script

```
<OP_2> <A pubkey> <B pubkey> <C pubkey> <OP_3>
OP_CHECKMULTISIG
```

NTNU

Micropayments: Historical View

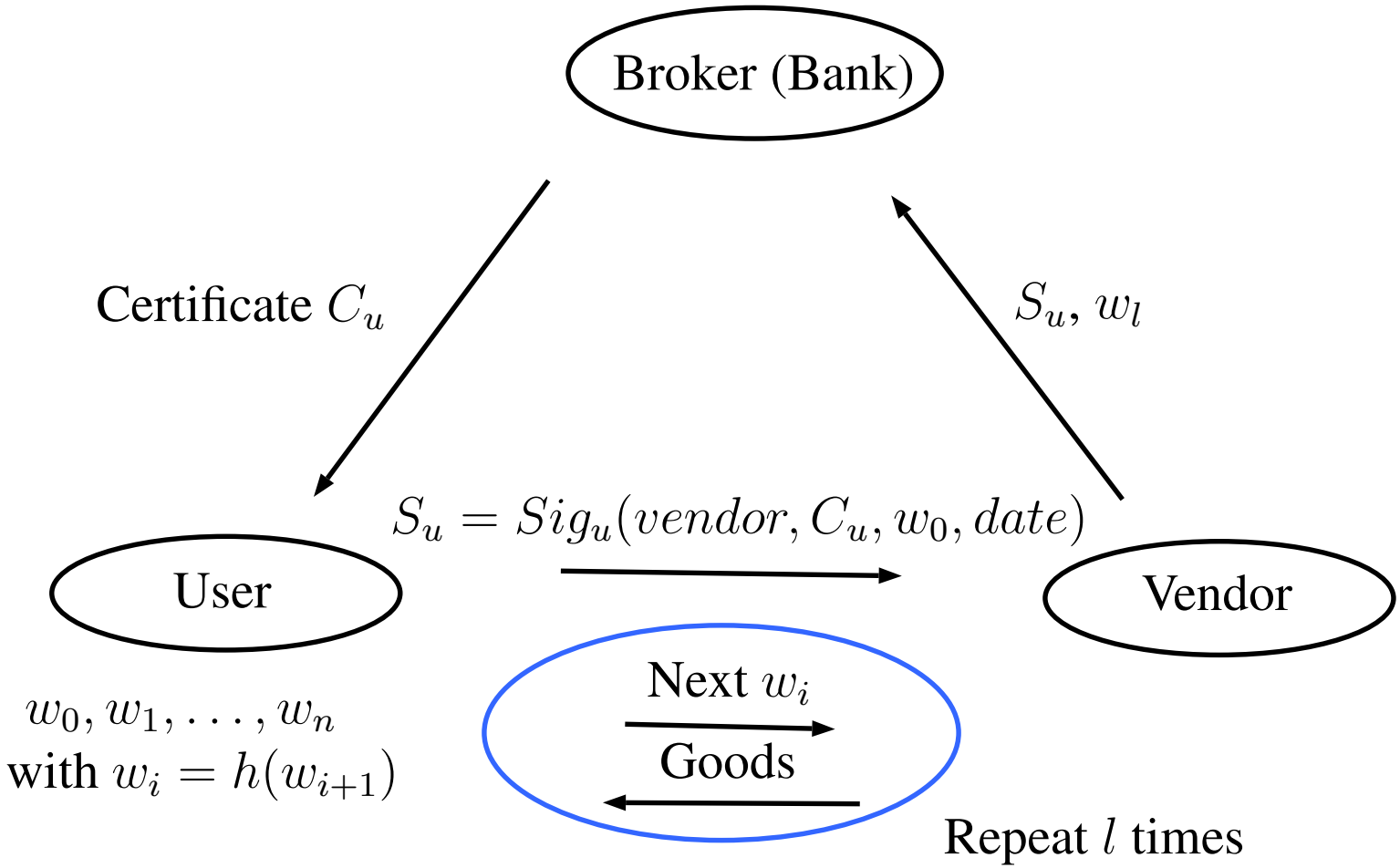
- 1990s when communication and computation were much more limited
- Problem that collecting payment is more than the goods are worth
- Minimise public key operations and communication with third parties
- Micromint and Payword (Rivest and Shamir, 1995)

NTNU

Payword

- Client registers with broker (bank) and obtains certificate C_u.
- Client constructs a *payword* using a reverse hashchain $w_0, w_1, \ldots, w_n$ with $w_i = h(w_{i+1})$
- User sends to vendor $Sig_u(vendor, C_u, w_0, date)$
- To purchase an item user sends next w_i
- One signature for many purchases
- Broker (bank) mostly offline

Payword


Broker (Bank)
Certificate C_u
S_u, w_l
$S_u = Sig_u(vendor, C_u, w_0, date)$
User
Vendor
$w_0, w_1, \ldots, w_n$
with $w_i = h(w_{i+1})$
Next w_i
Goods
Repeat l times

NTNU

Micromint

— A coin is a 4-way hash collision x_1, x_2, x_3, x_4 with

$$h(x_1) = h(x_2) = h(x_3) = h(x_4)$$

— Each month the coin issuer works to construct coins (minting)

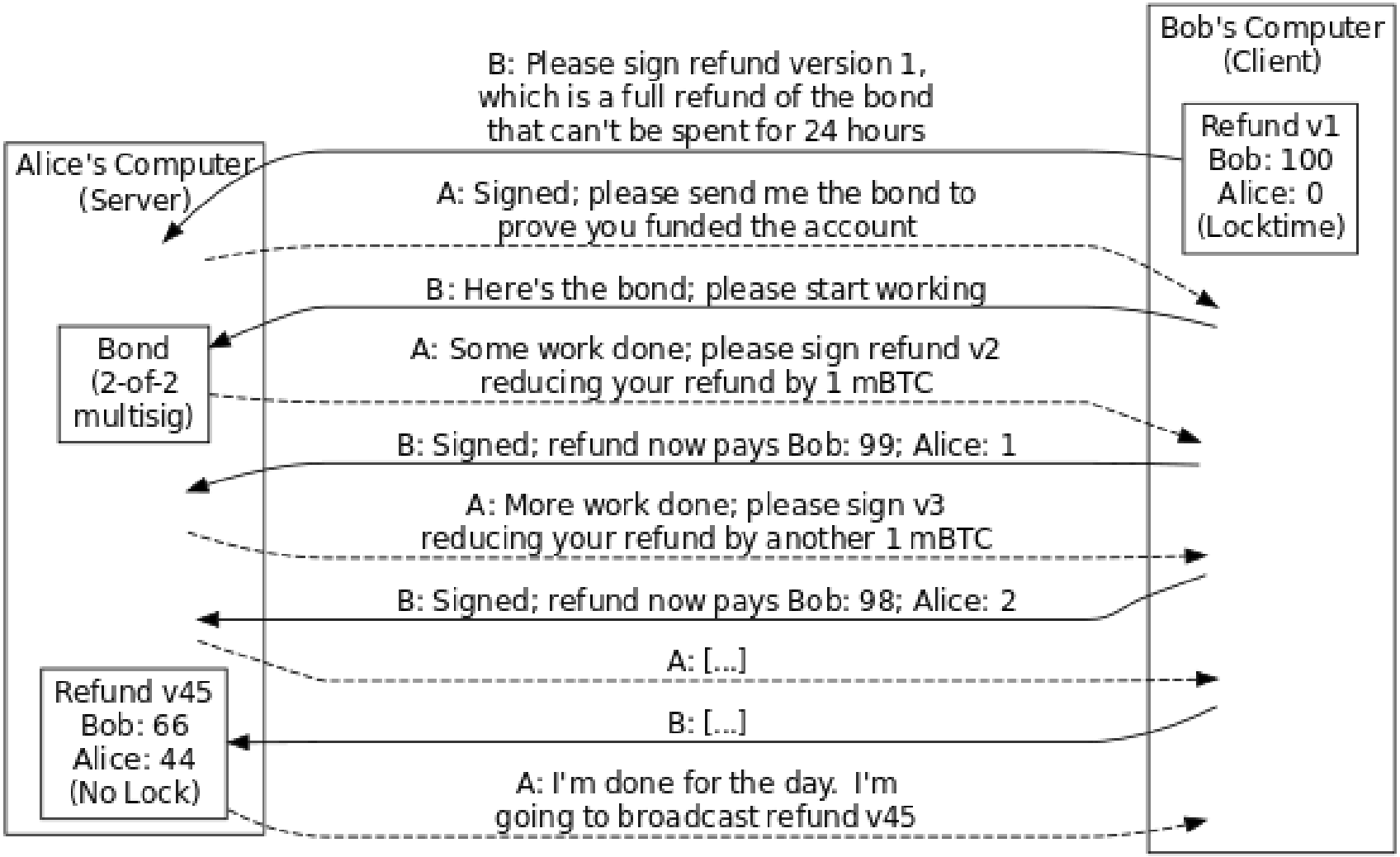
— New (tweaked) hash function each month

— Coins are cleared after they are spent

Micropayments: Bitcoin View

- Reduce number of payments in the blockchain
- Reduce transaction fees
- Problem that transaction fee is more than an individual transaction is worth
- Make offline protocol between a client and vendor, a *payment channel*
- Use of 2 out of 2 multisignatures
- Use *locktime* which prevents transaction to be executed until after a specific time or block height
- In next figure, the *bond* is a P2SH from Bob to address requiring multisignature of both Alice and Bob

Micropayment Channel (Source: `bitcoin.org`)



Micropayment Contract

— Only one transaction appears on the blockchain
— Only one transaction fee for multiple purchases
— Security analysis?
— Ideas expanded in the *Lightning Network* proposal

NTNU

Future

- Hundreds of bitcoin alternatives deployed today:
  - change genesis block to start your own Bitcoin version
  - change parameters
  - change protocol
- Making mining useful
- Alternatives to mining (proof of work)
- More complex contracts to trigger payments
- More anonymity, faster block times, provable security, different incentives, block graphs, off-chain payments, ...

Consensus

- Who decides what goes into the blockchain?
- Two types:
  - *permissionless blockchain*
  - *permissioned blockchain*
- Consensus: rules for agreeing what is a valid block

Other consensus mechanisms

- Proof of stake
- Sortition (Algorand)
- Byzantine agreement protocols on permissioned blockchains

NTNU

More information

- Narayanan, Bonneau, Felten, Miller, Goldfeder, Clark, Bitcoin and Cryptocurrency Technologies
  `http://bitcoinbook.cs.princeton.edu/`
- Joseph Bonneau overview slides:
  `http://jbonneau.com/presentations.html`
- Florian Tschorsch and Björn Scheurmann, Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies, IEEE Communications Surveys and Tutorials, 18, 3, 2016
- Technical details of Bitcoin: `en.bitcoin.it`
- Software and wallets for Bitcoin: `bitcoin.org`
- Live information and statistics: `blockchain.info` and `blockexplorer.com`
- Original Bitcoin paper of Satoshi Nakomoto:
  `https://bitcoin.org/en/bitcoin-paper`