Bitcoin,
a look under the bonnet
Outline

• Bitcoin 101
• Crypto
  – one-way functions
  – digital signatures
• Transactions
• Mining
• Binding it all together
• Anti-censorship
• Cautionary notes
What this talk is NOT about

NOT:

• History of cryptocurrencies
• Economics / politics
• How to become rich
• How to attack Bitcoin
Bitcoin 101

• Cryptocurrency
  – assets are purely digital
  – secret key gives access to "account"
  – crypto for proving ownership
  – crypto for signing transactions
  – crypto for creating new money

• Decentralized
  – peer to peer communication

• "Block chain"
  – all transaction history is public
Quick facts about Bitcoin

• Created in 2008 by "Satoshi Nakamoto"
  – open source
  – based on lots of prior work
  – but with unique combination of ingredients

• Accepted by $>10^5$ merchants
  – easy (international) transfer of money
  – PR value

• Current exchange rate:
  1 bitcoin = 209 euro
Who is Satoshi Nakamoto?

1 bitcoin = \(10^8\) Satoshi

Dorian S. Nakamoto?
(Los Angeles, March 2014)

There are many allegations
"Cryptographic hash function"

- easy to compute
- very difficult to invert
- low prob. of collisions
- compressed & unique digest

Example: SHA256

- arbitrary input size
- output size 256 bits
Proof of work

Given: some unpredictable stuff

Find out what to put here!

Hash value must satisfy a tough constraint, e.g. lie in a certain small range.

• Solving this problem costs a lot of trial & error => CPU time.
• Solutions are easily verified!
Concept #2: Digital signature

Main concept: key pair \((a, A)\)

- **Private key** (secret)
- **Public key**

\(A\) can be computed from \(a\); the reverse is difficult.

**Signing**

\(a\)

\(\text{random } r\)

\(\text{hash} \rightarrow h\)

\(S\)

\(\sigma = S(h, a, r)\)

**Signature**

**Signature verification**

\(V(h, \sigma, A) \in \{\text{yes, no}\}\)

\(A\)

\(\text{hash} \rightarrow h\)
Digital signatures

Anyone can verify a signature
– only the public key is needed

Only one person can generate signatures consistent with A
– private key a is needed

Successful verification of signature proves two things
1. message integrity
2. authenticity

"The holder of the private key confirms this message"
Choose your own key pair

• public key is your "account number"
• private key gives access to account
  – prove that the account is yours: signature
  – confirming a payment: sign it

You can make as many accounts as you want, completely for free!
**Bitcoin transactions**

**Transaction data structure**

- **In** data:
  - Payments to your account
  - Each "in" data proves ownership
    - Digital signature
    - Possibly different accounts
  - Each signature also covers all the "out" data

- Small transaction fee: \( \text{out} < \text{in} \)
The **Script** language is too expressive

- leads to implementation bugs
- many options are now forbidden
# Allowed challenge script types

<table>
<thead>
<tr>
<th>Type</th>
<th>Prove knowledge of:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pay to Pub.key</td>
<td>private key</td>
<td></td>
</tr>
<tr>
<td>Pay to Pub.key hash</td>
<td>pub.key &amp; priv.key</td>
<td></td>
</tr>
<tr>
<td>Multisig</td>
<td>n out of k private keys</td>
<td>n,k heavily restricted</td>
</tr>
<tr>
<td>Nulldata</td>
<td>---</td>
<td>40 bytes arbitrary data; Max one Nulldata per transaction</td>
</tr>
<tr>
<td>Pay to script hash</td>
<td>challenge script + whatever it demands</td>
<td>Must be one of the above script types</td>
</tr>
</tbody>
</table>
Mining

Users

We want to do these transactions:

peer-to-peer

Miners

• Collect transactions
• Signatures OK?
• Fees OK?
• Scripts OK?
• Double spending?
• Proof of work

BLOCK

pointer to previous block
list of collected transactions
coinbase transaction

Find out what to put here

hash

should be smaller than target $T$

$T \approx 2^{188}$

Lowered every 2016 blocks

(simplified view)
Incentives for the miner:
• reward for proof of work, currently 25 BC
• transaction fees

The incentives stabilize the system
• transaction confirmation mechanism brings reward
Anti-censorship

Whole block chain must be visible
• otherwise you cannot trust bitcoin
• Difficult to censor!

How to write data into blockchain?
• Control over data within transaction
• Plenty of options!

Krzysztof Okupski
MSc thesis, Dec. 2014:

"(Ab)using Bitcoin for an anti-censorship tool"
• Bitcoin specification document
• open-source tool
Embedding data in transactions

- Create $2^N$ accounts $\rightarrow$ N bits of info in account identifier
- Keep pumping ALL money through your accounts
- Pay to scripts, with 1-out-of-14 multisig

Put information in:
- nulldata output
- choice of 14 public keys in challenge scripts
- permutation of "in" w.r.t. "out"
- splitup of the budget (combinatorics: "composition")
- signatures

Additional tricks:
- text only
- reduced character set
- compression

Fees: 16 Satoshi per embedded byte
Reading is not dangerous

• Anybody can read the blockchain
• You don't even have to be part of the P2P network
• What you need to know:
  – where to look
  – how to parse
Remarks (1/2)

Black market, crime, etc.
• Public keys are pseudonyms
• But: full history is visible; exchange knows your identity
• Special effort needed for laundering/mixing/anonymizing

Theft
• ∃ specialized bitcoin-stealing malware
• Store private keys offline
• Use private keys offline (air gap)

Supposedly decentralized, but ...
• Small number of entities dominates mining
• Chinese mining farms
Bitcoin crypto seems flawless
• That's a first.

Bitcoin is cluttered
• scripts instead of fixed flowchart
• too many transaction types
• ....

Mining is a tremendous waste of energy
• Nothing useful is created, only "art"
• Find better confirmation mechanism
A look under Bitcoin's bonnet
• one-way hashes
• signed transactions
• proofs of work
• meshed together in the block chain

Well designed system basics
• stabilizing incentives
• solid crypto

Anti-censorship
• embed data in transactions
• cost = fees: ≈16 Satoshi/byte