SpaceMint
A cryptocurrency based on proofs of space

Georg Fuchsbauer

joint work with
S. Park, A. Kwon, K. Pietrzak, J. Alwen and P. Gaži

Inria Blockchain Meeting, 12/12/16
Bitcoin

- Digital currency
- Decentralized (no bank issuing coins)
- Pseudonymous
- Controled Inflation
Public ledger (maintained by authority)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
</tr>
</tbody>
</table>

Alice: transfer 1 → Bob
### Primer

#### Public ledger (maintained by authority)

<table>
<thead>
<tr>
<th></th>
<th>Alice</th>
<th>Bob</th>
<th>Charlie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Alice: transfer 1 → Bob
Primer

Public ledger (maintained by authority)

Alice 1
Bob 0
Charlie 0

Alice: transfer 1 → Bob

not private
Pseudonyms?

<table>
<thead>
<tr>
<th>Id</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id-0001</td>
<td>1</td>
</tr>
<tr>
<td>Id-0002</td>
<td>0</td>
</tr>
<tr>
<td>Id-0003</td>
<td>0</td>
</tr>
</tbody>
</table>

how to identify?
Digital signatures

- Alice can create a **key pair**
  - **private key** used to sign messages
  - **public key** lets anyone verify signatures
Digital signatures

- Alice can create a **key pair**
  - **private key** used to sign messages
  - **public key** lets anyone verify signatures

- **Unforgeability**: no one can forge signature w/o knowing secret key
Digital signatures

- Alice can create a **key pair**
  - **private key** used to sign messages
  - **public key** lets anyone verify signatures

- **Unforgeability**: no one can forge signature w/o knowing secret key

- Public key ≈ identity ≈ coin
- Private key: enables spending of coin
Transactions

- Alice owns $pk_A$ i.e. it’s in the ledger
- Bob creates $pk_B$
- Alice signs $pk_A \rightarrow pk_B$ and adds to ledger
Double-spending

- Alice signs $pk_A \rightarrow pk_B$
- Alice signs $pk_A \rightarrow pk_C$

Physical coin $\neq$ digital coin

hard to create

easy to copy!
Double-spending

- Alice signs $pk_A \rightarrow pk_B$
- Alice signs $pk_A \rightarrow pk_C$

Physical coin $\neq$ digital coin

Ledger only accepts if
- exists transaction $* \rightarrow pk_A$
- no transaction $pk_A \rightarrow *$
Decentralization

How to eliminate authority that

- checks validity of tx’s
- publishes list of tx’s (ledger)
Decentralization

How to eliminate authority that

- checks validity of tx’s
- publishes list of tx’s (ledger)

The Blockchain
Cryptographic hash functions
Cryptographic hash functions

Doc $\rightarrow$ \textbf{H} $\rightarrow$ Hash
Cryptographic hash functions

- outputs look random
  ⇒ small mods result in huge changes
  ⇒ hard to find preimage
Cryptographic hash functions

- outputs look random
  - small mods result in huge changes
  - hard to find preimage

  ⇒ best way to find input with hash from some subset is randomly trying
The Blockchain

- Blocks linked by including hash of previous block
  ⇒ cannot modify block w/o changing everything after
The Blockchain

- blocks linked by including hash of previous block
  ⇒ cannot modify block w/o changing everything after

acts as fingerprint for whole chain
The Blockchain

- transactions collected into block
- new block added & published every 10min
  ⇒ who adds block?
The Blockchain

- transactions collected into block
- new block added & published every 10 min
  ⇒ who adds block?
- assume mechanism chooses random user
  ⇒ user could be malicious
  ⇒ Sybil attacks?
  ⇒ Proof of work
Proof of work

• prove that you’ve performed work
• e.g. prevent spam: **Hashcash**
Proof of work

- prove that you’ve performed work
- e.g. prevent spam: **Hashcash**

\[ H \]

\[ 0...018730 \]

20×

random value

\[ \text{X-Hashcash: 0109161445:gfuchsba@inria.fr:0101} \]
Proof of work

- prove that you’ve performed work
- e.g. prevent spam: Hashcash

\[ X\text{-Hashcash: } 0109161445:gfuchsba@inria.fr:0101 \]

\[
\begin{array}{c}
\text{H} \\
\downarrow \\
0...018730 \\
\downarrow 20 \times
\end{array}
\]

- try out \( \approx 2^{20} \) values (\( \sim 1s \))
- easy to verify (\( \sim 1\mu s \))
Mining

\[ tx_1, tx_2, tx_3 \]

\[ h, $ \]
Mining

- collect transactions
- find value $ yielding small hash
Mining

- collect transactions
- find value $ yielding small hash
- broadcast block
Mining

- collect transactions
- find value $ yielding small hash
- broadcast block

\[
\begin{align*}
tx_1 & \\
tx_2 & \\
tx_3 & \\
\color{red}{h} & \\
\color{green}{$} & \\
\text{H} & \xrightarrow{65 \times} 0\ldots000730 \\
\end{align*}
\]

if
- tx’s are valid
- hash is small enough

⇒ add block to local copy of blockchain
Mining

- Incentive?
  ⇒ reward bitcoins!

(all bitcoins created this way)
Forks

$tx_1$
$tx_2$
$tx_3$
Forks

• Double-spending!
Forks

"Always mine on the longest chain"
Forks

“Always mine on the longest chain”
The “51%-attack”
The "51%-attack"

Secure if majority of miners is honest

⇒ wait for 6 blocks before accepting payment
Why does it work?
Why does it work?

- Miners incentivized by rewards
- Probability of mining block $\approx$ computing power
  $\Rightarrow$ no Sybil attacks!
- Rational to mine on longest chain
  $\Rightarrow$ quick consensus
Why does it work?

• Miners incentivized by rewards
• Probability of mining block $\approx$ computing power $
\Rightarrow$ no Sybil attacks!
• Rational to mine on longest chain $
\Rightarrow$ quick consensus

Problems
• specialized hardware / mining oligarchy
• Bitcoin consumes electricity like town of 100k population
Why does it work?

- Miners incentivized by rewards
- Probability of mining block $\approx$ computing power  
  $\Rightarrow$ no Sybil attacks!
- Rational to mine on longest chain  
  $\Rightarrow$ quick consensus

Problems

- specialized hardware / mining oligarchy
- Bitcoin consumes electricity like town of 100k population  
  $\Rightarrow$ Can proof of work be replaced by something else?
Proof of stake

- prob. of mining $\approx$ number of coins owned

- **Problems:**
  - *Nothing-at-stake problems*
  - *Participation: miners = holders*
Proof of space

- prove that you’ve allocated disk space

Trivial solution

Verifier       Prover
Proof of space

• prove that you’ve allocated disk space

Trivial solution

Initialization:

Verifier  \[\text{file } F\]  Prover
Proof of space

- prove that you’ve allocated disk space

**Trivial solution**

*Initialization:*

\[ \text{Verifier} \rightarrow \text{file } F \rightarrow \text{Prover} \]

*Prove:*

\[ \text{indices } i_1, \ldots, i_n \rightarrow F[i_1], \ldots, F[i_n] \]

- compare with \( F \)
Proof of space

- prove that you’ve allocated disk space

**Trivial solution**

**Initialization:**

Verifier \[\rightarrow\] Prover

\[\text{file } F\]

**Prove:**

- compare with \( F \)

\[ F[i_1], \ldots, F[i_n] \]

inefficient for verifier
Proof of space

- prove that you’ve allocated disk space

A better solution

**Initialization:**

Verifier | $f$ | Prover

- store lookup table
  - $(1, f(1))$
  - $\vdots$
  - $(N, f(N))$
- sorted by output
Proof of space

- prove that you’ve allocated disk space

A better solution

Initialization:

Verifier \[ f \] Prover

\[ y_1, \ldots, y_n \]

\[ \pi = (f^{-1}(y_1), \ldots, f^{-1}(y_n)) \]

Prove:

- store lookup table
  \[ (1, f(1)) \]
  
  \[ \vdots \]

  \[ (N, f(N)) \]

sorted by output
Proof of space

- prove that you’ve allocated disk space

A better solution

**Initialization:**

Verifier \[ f \]

Prover

- store lookup table \((1, f(1))\)

\[ \pi = (f(1), \ldots, f(n)) \]

**Proof:**

Time/memory trade-offs:

Store \(N^{2/3}\), invert in time \(N^{2/3}\)

\[ \Rightarrow \] Proof of work
Proof of space

- prove that you’ve allocated disk space

[DFKP’15]

Initialization:

Verifier \( id \) Prover
Proof of space

- prove that you’ve allocated disk space

[DFKP’15]

**Initialization:**

Verifier \(\rightarrow id\) \(\rightarrow Prover\)

- fill nodes of graph \(G\) dep. on \(id\)
- hash content
Proof of space

• prove that you’ve allocated disk space

[DFKP’15]

*Initialization:*

Verifier  \[id\]  Prover

- fill nodes of graph \(G\) dep. on \(id\)
- hash content

(use hard-to-pebble graph)
(hash using Merkle tree)
Proof of space

- prove that you’ve allocated disk space

[DFKP’15]

**Initialization:**

Verifier

\[ \text{id} \]

Prover

\[ \gamma \]

- fill nodes of graph \( G \) dep. on \( \text{id} \)
- hash content

**Prove:**

- check consistency with \( \gamma \)

\[ \pi = (G[i_1], \ldots, G[i_n]) \]
SpaceMint

- replace proof of **work** by proof of **space**

- **Advantages:**
  - *green:* low electricity; reusable hardware
  - *decentralized*
SpaceMint

• replace proof of work by proof of space

• Advantages:
  – green: low electricity; reusable hardware
  – decentralized

• Challenges:
  – PoS is interactive
  – Nothing-at-stake problems
SpaceMint

• replace proof of **work** by proof of **space**

• **Advantages:**
  – **green:** low electricity; reusable hardware
  – **decentralized**

• **Challenges:**
  – PoS is *interactive*
  – *Nothing-at-stake problems*
    * Mining multiple chains
    * Grinding blocks
Miner initializes space with $id = pk$
- broadcasts $\gamma$
- $\gamma$ gets added to chain
proof $\pi$ for hash $\gamma$ with challenge $c$
Who gets to add the block?
Who gets to add the block?

- Quality of block?
  ⇒ define fct. of proof $\pi$: quality $\sim$ space allocated
- Block with best proof gets added to chain
- Blocks define quality of chain
SpaceMint

\(tx_1\) \(tx_2\) \(tx_3(\gamma)\)

\[\neq \text{Bitcoin: } 1\]

- easy to check if good solution!
  \[\Rightarrow \text{ miners try to extend every chain}\]
  \[\Rightarrow \text{ no consensus!}\]
SpaceMint

$tx_1$
$tx_2$
$tx_3(\gamma)$

$\neq$ Bitcoin: ①

● easy to check if good solution!

⇒ miners try to extend every chain

⇒ no consensus!

Forbid extending 2 chains
SpaceMint

≠ Bitcoin: ②

- easy to check if good solution!
  ⇒ miners might not extend best chain
  ⇒ no consensus!
SpaceMint

$tx_1$
$tx_2$
$tx_3(\gamma)$

$\neq$ Bitcoin: $\boxed{2}$

- *easy* to check if good solution!

  => *miners might not extend best chain*

  => *no consensus!*

*Take challenge from past*
SpaceMint

- easy to check if good solution!

⇒ miners might not extend best chain
⇒ no consensus!

Take challenge from past
SpaceMint

\[ \neq \text{Bitcoin: } 2 \]

- easy to check if good solution!
  - miners might not extend best chain
  - no consensus!

Take challenge from past
SpaceMint

\( \neq \text{Bitcoin: } 3 \)

\( \Rightarrow \text{miners might grind blocks leading to good challenge in future} \)

\( \Rightarrow \text{proof of work} \)
Bitcoin: ⇒ miners might grind blocks leading to good challenge in future

⇒ proof of work

Make challenge hash of $\pi$ only
SpaceMint

$tx_1$
$tx_2$

$\pi$

$\neq$ Bitcoin: 3

$\Rightarrow$ miners might grind blocks leading to good challenge in future

$\Rightarrow$ proof of work

Make challenge hash of $\pi$ only
• Transactions not hashed
  ⇒ not consolidated in chain!

• Blocks not linked to previous block
  ⇒ consensus??
SpaceMint

- Transactions not hashed
  ⇒ not consolidated in chain!
- Blocks not linked to previous block
  ⇒ consensus??

New blockchain structure
SpaceMint
Use **signatures** (tied to proof) to link blocks
Use signatures (tied to proof) to link blocks
More ecological?

- no ongoing cost
- resources recyclable
- unused disk space $\Rightarrow$ decentralized
Y a-t-il des questions?