An analysis of the applicability of blockchain to secure IP addresses allocation, delegation and bindings

draft-paillisse-sidrops-blockchain-00

IETF 99 - Prague
July 2017

Jordi Paillissé, Albert Cabellos, Vina Ermagan, Alberto Rodríguez, Fabio Maino
jordip@ac.upc.edu

http://openoverlayrouter.org
A short Blockchain tutorial
Blockchain - Introduction

• Blockchain:
  – Decentralized, secure and trustless database
  – Token tracking system (who has what)

• Add blocks of data one after another
• Protected by two mechanisms:
  – Chain of signatures
  – Consensus algorithm

• First appeared: Bitcoin, to exchange money
• Other applications are possible
Blockchain - Transactions

Transaction
  Sender's Public Key
  Sender's signature
  Data
Blockchain - Transactions

1. Transactions are broadcasted to all the nodes

<table>
<thead>
<tr>
<th>Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender's Public Key</td>
</tr>
<tr>
<td>Sender's signature</td>
</tr>
<tr>
<td>Data</td>
</tr>
</tbody>
</table>

P2P network
Blockchain - Transactions

1. Transactions are broadcasted to all the nodes

2. A node collects transactions into a block

Block
- Previous Hash
- Transactions 1 \cdots N
Blockchain - Transactions

1. Transactions are broadcasted to all the nodes

2. A node collects transactions into a block

3. Compute consensus algorithm

Transaction
- Sender's Public Key
- Sender's signature
- Data

Transactions 1’ ··· N’

Previous Hash

Transactions 1 ··· N

Previous Hash

Transactions 1’ ··· N’
Blockchain - Transactions

1. Transactions are broadcasted to all the nodes

2. A node collects transactions into a block

3. Compute consensus algorithm

4. Broadcast new block to the network
Blockchain - Transactions

1. Transactions are broadcasted to all the nodes

2. A node collects transactions into a block

3. Compute consensus algorithm

4. Broadcast new block to the network

5. The other nodes verify the consensus algorithm and accept the block
Blockchain - Properties

- Decentralized: all nodes have the entire blockchain
- No prior trust required
- Decouples ownership from identity
- Append-only and immutable: added transactions cannot be modified
- Verifiable
### Chain of signatures

<table>
<thead>
<tr>
<th>Sender A</th>
<th>Data</th>
<th>Receiver B</th>
</tr>
</thead>
<tbody>
<tr>
<td>P+A</td>
<td>Sign (P+A)</td>
<td>Token #123</td>
</tr>
</tbody>
</table>

Only the owner of P-B can send this token.

<table>
<thead>
<tr>
<th>Sender B</th>
<th>Data</th>
<th>Receiver C</th>
</tr>
</thead>
<tbody>
<tr>
<td>P+B</td>
<td>Sign (P+B)</td>
<td>Token #123</td>
</tr>
</tbody>
</table>
## Chain of signatures

<table>
<thead>
<tr>
<th>Sender A</th>
<th>Data</th>
<th>Receiver B</th>
</tr>
</thead>
<tbody>
<tr>
<td>P+A</td>
<td>Sign (P+A)</td>
<td>Token #123</td>
</tr>
</tbody>
</table>

Only the owner of P-B can send this token.

<table>
<thead>
<tr>
<th>Sender B</th>
<th>Data</th>
<th>Receiver C</th>
</tr>
</thead>
<tbody>
<tr>
<td>P+B</td>
<td>Sign (P+B)</td>
<td>Token #123</td>
</tr>
</tbody>
</table>

Add it again → impossible.
Consensus algorithm

- Central part of blockchains
- Controls addition of blocks
- Defines what is consensus
- Most common:
  - Proof of Work, e.g. Bitcoin
  - Proof of Stake, e.g. Ethereum (shorty)
Proof of Work

• Perform a large number of calculations
• Eg: find nonce so that:

$$\text{SHA-256 (transactions + hash (prev. Block) + nonce)} = 00000000xxxxxxxxxxxxxxx$$

• Change data $\Rightarrow$ redo Proof of Work
• Accumulate computing power

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>...</th>
<th>300</th>
<th>301</th>
<th>302</th>
<th>303</th>
<th>Last block</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>impossible</td>
<td>hard</td>
<td>easy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Not necessarily performed by the users of the blockchain
Proof of Stake

- Any owner of tokens can add a block
- Selected randomly
- Users with more tokens are more likely to be selected
  - Reduced incentive to attack (because they use the blockchain)
- Attacks are different than PoW
Proof of Stake

List all stake

<table>
<thead>
<tr>
<th>Holder</th>
<th># tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>124</td>
</tr>
<tr>
<td>E</td>
<td>110</td>
</tr>
<tr>
<td>B</td>
<td>87</td>
</tr>
<tr>
<td>D</td>
<td>75</td>
</tr>
<tr>
<td>F</td>
<td>54</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>
Proof of Stake

List all stake

<table>
<thead>
<tr>
<th>Holder</th>
<th># tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>124</td>
</tr>
<tr>
<td>E</td>
<td>110</td>
</tr>
<tr>
<td>B</td>
<td>87</td>
</tr>
<tr>
<td>D</td>
<td>75</td>
</tr>
<tr>
<td>F</td>
<td>54</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>

Sign new block (E priv. key)

Weighted random selection
Summary of features

vs. traditional PKI systems

Advantages
• Decentralized
• No CAs
• Simplified management
• Simple rekeying
• Limited prior trust
• Auditable
• Censorship-resistant

Drawbacks
• No crypto guarantees
• Large storage
• Costly bootstrapping
Blockchain for IP addresses
Data in the blockchain

We want to store:

- Prefix: 10/8  
  Holder: P+

- IP address block + Holder

- Prefix: 10/8  
  Holder: P1+

- Prefix: 10/8  
  Holder: P2+

- Prefix: 10/8  
  Holder: P3+

- Prefix: 10/8  
  AS#: 12345

Chain of allocations and delegations
IP addresses vs. coins

• IP addresses = coins

• Similar properties:
  – Unique
  – Transferrable
  – Divisible

• Exchange blocks of IP addresses just like coins
Which consensus algorithm?

• PoW presents some drawbacks:
  – Parties that add blocks do not necessarily use the blockchain
  – Takeover if enough computing power
  – Hardware dependency
  – Energy inefficiency

https://www.bitcoinmining.com/
Which consensus algorithm?

- PoS appears to be more suitable for this scenario:
  - No special hardware
  - No expensive computations
  - Parties with more IP addresses control the blockchain
  - Users of the blockchain maintain it
Why Proof of Stake?

• PoS appears to be more suitable for this scenario:
  – Takeover requires accumulating a large amount of IP blocks
  – Participants do not have an incentive to sell IP blocks to an attacker
Example
From: IANA
To: IANA
I have all prefixes
From: IANA
To: IANA
I have all prefixes

From: IANA
To: APNIC
Prefix 1/8 for APNIC
From: IANA  
To: IANA  
I have all prefixes

From: IANA  
To: APNIC  
Prefix 1/8 for APNIC

From: APNIC  
To: ISP A  
ISP A has 1.2/16
From: IANA
To: IANA
I have all prefixes

From: IANA
To: APNIC
Prefix 1/8 for APNIC

From: APNIC
To: ISP A
ISP A has 1.2/16

From: ISP A
To: ISP A
Bind 1.2/16 to AS # 12345
Who has 1.2/16?

From: IANA
To: IANA
I have all prefixes

From: IANA
To: APNIC
Prefix 1/8 for APNIC

From: APNIC
To: ISP A
ISP A has 1.2/16

From: ISP A
To: ISP A
Bind 1.2/16 to AS # 12345

From: ISP A
To: ISP A
Bind 1.2/16 to AS # 12345

AS# 12345
Who has 1.2/16?

AS# 12345

I can go back to check if this prefix was originally owned by IANA
Our use case

- LISP Beta Network
- Uses LISP-DDT*
- Full mapping system in the blockchain

*http://ddt-root.org/*
Thanks for listening!
Scalability

- One AS <> prefix binding for each block of /24 IPv4 address space
- Growth similar to BGP churn*
- Each transaction approx. 400 bytes
- Only IP Prefixes: worst case + BGP table growth*: approx. 40 GB in 20 years
- With PoS, storage can be reduced

*Source: http://www.potaroo.net/ispcol/2017-01/bgp2016.html

Approx. 600 GB in 2034 (IP blocks + AS bindings)
Transaction examples
First transaction

- Users trust the Public Key of the Root, that initially claims all address space by writing the genesis block.
- Root can delegate all address space to itself and use a different keypair.

\[
\text{Hash}(P + \text{root}) = \text{Root@1} \quad \text{New Transaction} \quad \text{“I own all the address space”} \quad \text{Root@2}
\]
Prefix allocation and delegation

- Root allocates blocks of addresses to other entities (identified by Hash(Public Key)) by adding transactions

  - Holders can further delegate address blocks to other entities
Writing AS bindings

• Just like delegating a prefix, but instead of the new holder, we write the binding

Deleg3@ \rightarrow \text{“binding”} \rightarrow 0.0.1/24 \text{ from AS# 12345}
Rekeying

• Delegating the block of addresses to itself using a new key set.

• Simpler than traditional rekeying schemes

• Can be performed independently, i.e. each holder can do it without affecting other holder

• Same procedure for AS number bindings
External server authentication

• Some information may not be suitable for the blockchain, or changes so fast it is already outdated when added into a block

• A public key from an external server can also be included in the delegations

• Since blockchain provides authentication and integrity for this key, parties can use it to authenticate responses from the external server
FAQ

• Does it grow indefinitely?
  – Yes
• Do all nodes have the same information?
  – Yes
• When answering a query, do you have to search the entire blockchain?
  – No, you can create a separate data structure only with the current data
• If I lose my private key, do I lose my prefixes also?
  – Yes, watch out!