Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2018

J. Paillisse
UPC-BarcelonaTech
A. Rodriguez-Natal
V. Ermagan
F. Maino
Cisco Systems
A. Cabellos
UPC-BarcelonaTech

July 03, 2017

An analysis of the applicability of blockchain to secure IP addresses allocation, delegation and bindings.
draft-paillisse-sidrops-blockchain-00.txt

Abstract

This document analyzes how blockchain technology can be used to secure the allocation, delegation and binding to topological information of the IP address space. The main outcomes of the analysis is that blockchain is suitable in environments with multiple distrusting parties and that Proof of Stake is a potential candidate for a consensus algorithm.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Blockchain [Bitcoin] is attracting a lot of attention among the security community since it provides means for exchanging information among a set of distrusting entities without the use of digital certificates and centralized control. Blockchain provides means for

the distrusting parties to reach consensus. Formally, it is regarded as a new solution to the Byzantine Generals problem, well-known in fault-tolerant distributed systems.

Although at the time of this writing the main application of blockchain are financial systems, their use in the field of networking is being explored (e.g., [Hari2016]). Some successful systems exist such as [Blockstack] and [Namecoin], which aim at building a secure DNS.

The main goal of this document is to represent a first step towards the understanding of the properties of blockchains and their applicability in the Internet infrastructure, specifically securing the allocation, delegation and bindings of IP addresses. First, it introduces blockchain, then it analyzes how blockchain could be used to secure the delegation of IP addresses. Finally, it presents an initial design for such an infrastructure. This document also includes a preliminary security analysis of such system. It is important to note that the goal of this document is not to provide a complete architecture that secures IP address allocation, delegation and bindings.

2. Definition of Terms

TBC

3. Blockchain in a nutshell

3.1. Overview

Conceptually, a blockchain is a distributed, secure and trustless database. It can also be regarded as a state machine with rules that clearly state which transitions can be performed. Participants in the blockchain communicate through a P2P network. The smallest data unit of a blockchain is a transaction. Users attach data to a transaction along with its signature and their associated public key. Usually, the attached data is an asset or a token, something that is unique and should not be replicated (e.g., coins in Bitcoin). Then they broadcast this transaction to the other participants. The rest of the nodes in the network store temporarily this transaction. At some fixed intervals in time, one of the nodes takes a set of these transactions and groups them in a block. It then broadcasts this block back to the network. When the other nodes receive this block they verify it, remove the transactions contained in the block from the temporary storage and add it after the previous block, thus creating a chain of blocks. It should be noted that all nodes store the entire blockchain locally. In addition, most blockchains give some sort of reward to nodes that add new blocks, although this is

not strictly necessary. Figure 1 presents an overview of the most common elements in a block.

```
+--------+--------------+---------------+--------------+
| Block  | Hash(Previous | Hash(All Block | Block Creator |
| Number |     Block)    | Transactions)  |   Signature   |
+--------+--------------+---------------+--------------+
|                      Transaction 1                      |
+------------------------------------------------------+
|                      Transaction 2                      |
+------------------------------------------------------+
+                           ...                           +
+                           ...                           +
+------------------------------------------------------+
|                      Transaction N                      |
+------------------------------------------------------+
```

Figure 1.- Common structure of a block

Two basic mechanisms are used to protect the chained data: a chain of signatures and a consensus algorithm.

3.1.1. Chain of signatures

The chain of signatures operates at transaction level. Consider the sender and receiver of a token, each with its public-private keypair. To change the owner of a token, the sender signs the data and the receiver's public key. It then puts together its public key, the signature, the data and the hash of the receiver's public key (Figure 2) to form a transaction.

```
+-------------+------------------+--------+--------------+
|   Sender    |  Signature Sender |  Data  | Hash(Receiver |
| Public Key  |    Private Key    |        |  Public Key)  |
+-------------+------------------+--------+--------------+
```

Figure 2.- Common transaction structure in a blockchain

In conclusion, the rules of the blockchain enforce that:

- The owner of the receiver private key has total control over the contents of the transaction. In Bitcoin this translates in a central property: only this owner can spend a coin.

- When an owner sends a token to the new owner, it irreversibly transfers the control of the contents to the new owner.

3.1.2. Consensus algorithm

The consensus algorithm is the central part of blockchain and it controls the chaining of data blocks. The main role of the algorithm is to provide a set of well-defined rules so that participants agree on a consistent view of the database. For this it has the following main functions. First, forks (multiple chains) can exist, this may happen for instance due to varying network latency among participants. In this case the participants must agree on which is the valid chain. And second, another important function of the consensus algorithm is to determine which participants are allowed to add a new data blocks. Section 3.3 contains more information regarding available consensus algorithms.

It is important to note that regardless of the consensus algorithm, in blockchain data blocks are always added, never deleted nor modified. This creates a tamper-proof, shared ledger among all participants. Transactions can be tracked back by inspecting past blocks, thus enabling the verification of claims by certain parties.

3.2. Features

The following list tries to briefly summarize the main characteristics of the blockchain technology:

Decentralized: No central entity controls the blockchain, it is shared among all participants.

No CAs: No digital certificates, Certification Authorities or CRLs are needed.

Limited prior trust: It is not required to trust other nodes. It is worth noting that some consensus algorithms rely on some limited levels of trust.

Tamper-proof: Since data can be only added but never modified, attempts to alter previous records are detected.

Non-repudiation: All nodes share a common, immutable view on the status of the blockchain, and blockchain provides non-repudiation mechanisms.

Censorship-resistant: Gaining control over a transaction involves having access to the associated private key.

Append-only: Data is always added, but never modified nor deleted.

Privacy: Entities participating in the blockchain can achieve privacy using anonymous keys, i.e. randomly-generated keys not related to their identity. In addition, a new keypair should be generated for each new transaction in order to prevent tracking [Bitcoin], section 10.

Slow updates: New transactions have to be verified, added to a block and received by all nodes. This results in a delay since the transaction is created until it is finally available to all the nodes. This delay will depend on the consensus algorithm and the block creation rate.

Large storage: The size of the blockchain keeps growing forever, because data blocks are always added. This may result in scalability issues.

3.3. Description of consensus algorithms

The two more popular consensus algorithms are: Proof of Work and Proof of Stake.

3.3.1. Proof of Work (PoW)

In Proof of Work nodes have to solve a complex mathematical problem to add a block, thus requiring some computational effort, this is commonly know as mining. For example in Bitcoin the problem is to find a hash starting with a fixed amount of zeroes, the only known way to solve this problem is by brute force. The valid chain is the one with most accumulated computing power, this chain is also the more expensive in terms of computing power to modify. This is because modifying a block going N blocks back from the tip of the chain would require redoing the computations for all these N blocks. As a result, an attacker should have more computational power than the power required to create the N blocks to be able to modify the chain. Overall, it is commonly assumed that if more than half of the nodes are honest the blockchain is considered as secure.

PoW offers relevant features, adding new blocks requires an external resource (CPU power) that has an economical cost. However this also results in some relevant drawbacks:

Risk of overtaking: The security of PoW is entirely based on computation power. This means that if an entity has access to

more than half of the total blockchain's computing power it can control the chain. As a result and in order to keep blockchain secure, the incentive of taking control of the chain must be lower than the cost of acquiring and operating the hardware that provides the equivalent to half of the participants computing power. This is hard to guarantee since the economy of the blockchain and the economy of the required hardware are independent. As an example an attacker can acquire the required hardware and operate it, take control of the blockchain to obtain an economical benefit and finally sell the hardware to reduce the final cost of the attack.

Hardware dependency: Bitcoin automatically increases -over time- the complexity of the mathematical problem that needs to be solved in order to add a block. This is done to account for Moore's law. As a result the community has designed mining specific hardware (ASICs) that provides a competitive advantage. In this context blockchain becomes less democratic, since the cost of participating in it increases.

Energy inefficiency: PoW requires large amounts of energy to perform the computations (e.g., [miningfarm]).

3.3.2. Proof of Stake (PoS)

The main idea behind Proof of Stake is that participants with more assets (or stake) in the blockchain are more likely to add blocks. With this, the control of the chain is given to entities who own more stake. For each new block, a signer is selected randomly from the list of participants typically weighted according to their stake. A fundamental assumption behind PoS is that such entities have more incentives for honest behaviour since they have more assets in the chain.

Proof of Stake is seen as an alternative to PoW. At the time of this writing major players in the blockchain environment such as [Ethereum] are preparing a shift towards PoS, moreover several blockchains based on PoS already exist (eg. [Peercoin]). The main reason behind this paradigm shift is that PoS addresses some of PoW's main drawbacks:

o It does not require special hardware nor computationally or energy-expensive calculations.

o An attacker must get hold of a significant part of the assets in order to gain control of the blockchain. As opposed to PoS the investment required to gain control of the chain lies within the chain, and does not involve using external resources.

On the other side, Proof of Stake introduces new sources of attacks:

- In Proof of Stake the signer is selected randomly among the stakers. In this context attackers can manipulate the source of randomness to sign more blocks and ultimately gain control over the chain.

- As opposed to PoW, creating forks is very inexpensive, since no computational power is required. The PoS must provide means to select the valid chain, which is typically the longer one.

- Collusions of high-stakers can create alternate chains which can appear to be valid.

4. Blockchain for IP addresses

4.1. Problem statement

The objective of this section is to analyze if an infrastructure using blockchain can provide a similar degree of security to traditional PKI-based architectures. Specifically we aim to secure:

- Binding of IP address blocks to the holder (private key holder).

- The allocations and delegations of IP address blocks among their holders.

- Binding of IP address blocks to their topological locators (eg. AS numbers allocations).

This information is public and shared among a set of distrusting entities over the Internet. The architecture must be able to:

- Allow anyone to verify the legitimate holder of a block of addresses

- Let participating entities allocate address blocks without requiring a trusted third party.

- Restrict the allocation of a block of addresses to only its legitimate holder.

- Prevent data modification without the consent of its holder.

4.2. Analysis

The main rationale behind using blockchain to secure IP address allocations is that IPs can be understood as coins, both concepts share some fundamental characteristics:

- They are unambiguously allocated to entities.
- Can be transferred between them.
- Cannot be assigned to two entities at the same time.
- Can be divided up to a certain limit.

Such similar properties make it possible to envisage a blockchain that allows its participants delegate IP address blocks, similarly to how Bitcoin transfers coins. For example, IANA could write a transaction allocating addresses to the RIRs, which in turn could allocate them to the LIRs, etc. Complex management logic can be defined as needed for example, rejecting a transaction that allocates of a block of addresses originated by an entity that does not hold that block. In addition, transactions accept multiple inputs and outputs, i.e. an arbitrary amount of public keys either as senders or receivers. This means that it is possible to break and merge blocks of addresses as required. Section 5 provides more detailed information about this architecture.

4.3. A consensus algorithm for IP addresses

As stated before, the consensus algorithm is a central part of a blockchain. The first consensus algorithm designed for blockchain was PoW, and it is a common choice for new blockchain implementations. However it presents several drawbacks (Section 3.3.1) for the IP address scenario.

Using computing power as a means to secure blockchains has been proved to work in financial environments. However, the capability to add new blocks and the security of the chain itself depends on the computing power of the participants, which is not always aligned with their interest in the well-being of the blockchain. Depending on the objectives of the attacker, certain attacks can become profitable. Namely, buying a large quantity of hardware to be able to rewrite the blockchain with false data (e.g., incorrect delegations of IP addresses). This is because the incentives of the participants of the IP addresses blockchain are not linked with their computing power.

In contrast, with Proof of Stake the capability to alter the blockchain remains within it. This aspect is of particular importance in the context of securing IP addresses: it would mean that AS domains holding large blocks of IP addresses are more likely to add blocks. These parties have a reduced incentive in tampering the blockchain because they would suffer the consequences: an insecure Internet. Typically ASes that hold large blocks of IP address space have their business within the Internet and as such, have clear incentives in the correct operation and security of the Internet.

Furthermore, in such blockchain the risk of takeover is reduced compared to PoW, the reason is that accumulating a large amount of IP addresses is typically more complex than accumulating computing power. The risk of takeover is also mitigated compared to other PoS-based blockchains. In this blockchain an attacker would buy tokens from the other parties, who receive a monetary compensation to participate in the attack. However, in a blockchain for IP addresses this would mean buying IP addresses from other parties, who do not have a clear incentive to sell their blocks of addresses to the attacker. Because of this, PoS appears to be a firm candidate for a consensus algorithm in a blockchain for securing IP addresses allocations and delegations.

5. Architecture overview

This architecture mimics the hierarchy of IP address allocation present in today's Internet, with IANA on top of it. All nodes trust IANA's public key, which writes a genesis transaction assigning all of the address space to itself (figure 3).

```
+--------------+-----------------+-----------+----------------+
|    IANA      | Signature IANA  |  Allocate |   Hash(IANA    |
| Public Key 1 |  Private Key 1  |  0/0      |  Public Key 2) |
+--------------+-----------------+-----------+----------------+
```

Figure 3.- Genesis transaction

It then begins allocating each block of addresses to the IP address holders. Each transaction allocates part of the address space to the legitimate holder, and the rest of space is given back to IANA using a new keypair (figure 4).

```
+--------------+-----------------+-----------+----------------+
|              |                 |  Rest of  |   Hash(IANA    |
|     IANA     | Signature IANA  |  space    |  Public Key 3) |
| Public Key 2 |  Private Key 2  +-----------+----------------+
|              |                 |  Allocate |   Hash(APNIC   |
|              |                 |  001/8    |  Public Key 1) |
+--------------+-----------------+-----------+----------------+
```

Figure 4.- Example allocation transaction

In turn, all the parties in the hierarchy allocate or delegate address blocks following the current allocation hierarchy. When a party wants to verify the allocation of a block of addresses, it downloads the blockchain and verifies all the blocks and transactions up to the genesis block, for which it has trust. Figure 5 presents an example of allocation of one prefix to each of the RIRs.

```
+--------------+-----------------+-----------+----------------+
|              |                 |  Rest of  |   Hash(IANA    |
|              |                 |  space    |  Public Key 4) |
|              |                 +-----------+----------------+
|              |                 |  Allocate |   Hash(RIPE    |
|              |                 |  005/8    |  Public Key 1) |
|              |                 +-----------+----------------+
|              |                 |  Allocate |   Hash(APNIC   |
|     IANA     | Signature IANA  |  014/8    |  Public Key 2) |
| Public Key 3 |  Private Key 3  +-----------+----------------+
|              |                 |  Allocate |   Hash(ARIN    |
|              |                 |  023/8    |  Public Key 1) |
|              |                 +-----------+----------------+
|              |                 |  Allocate |   Hash(AFRINIC |
|              |                 |  102/8    |  Public Key 1) |
|              |                 +-----------+----------------+
|              |                 |  Allocate |   Hash(LACNIC  |
|              |                 |  200/8    |  Public Key 1) |
+--------------+-----------------+-----------+----------------+
```

Figure 5.- Example multi-output allocation transaction

Inside the blockchain the typical operations to manage blocks of IP addresses can be defined, such as the delegation of prefixes (figure 6). This helps to enforce the rules of IP addresses management. For instance, since this transaction is marked as a delegation, if the

new owner created an allocation transaction it would be rejected by the other nodes, because the parent transaction does not have the privileges to perform it.

```
+--------------+-----------------+-----------+----------------+
|              |                 | Rest of   |   Hash(APNIC   |
|    APNIC     | Signature APNIC | space     |  Public Key 3) |
| Public Key 1 |  Private Key 1  +-----------+----------------+
|              |                 | Delegate  |  Hash(Big ISP  |
|              |                 | 001.002/16|  Public Key 1) |
+--------------+-----------------+-----------+----------------+
```

Figure 6.- Example delegation transaction

This chain can define as many operations as required, for instance storing the binding of AS numbers to the IP prefixes they announce (figure 7).

```
+--------------+-----------------+-----------+----------------+
|              |    Signature    | Bind      |                |
|   Big ISP    |     Big ISP     | 001.002/16|  Hash(Big ISP  |
| Public Key 1 |  Private Key 1  | AS no.    |  Public Key 2) |
|              |                 | 12345     |                |
+--------------+-----------------+-----------+----------------+
```

Figure 7.- Example binding of AS number to prefix

Additional and more complex operations can be defined if the management logic requires it. For instance, several signatures (from different parties) can be required to consider a transaction valid, etc.

5.1. Pros and cons

In this section we analyze the pros and cons of a PoS-based blockchain system compared to traditional PKI infraestructures:

Advantages:

- Decentralized: No central entity controls the blockchain, it is shared among all participants.

- No CAs, CRLs or certificates needed: No digital certificates, Certification Authorities or CRLs are needed.

- Simplified rekeying: A key rollover can easily be performed by issuing a new transaction allocating the prefixes to a new keypair controlled by the same holder. This process can be performed without involving any third-party.

- Censorship-resistant: since the control of a transaction is completely under the holder of the private key, the revocation of IP addresses without the legitimate holder's permission involves obtaining its private key. Even if the private key of the previous owner was compromised, ownership of the current transaction is still preserved, as opposed to the compromise of a CA's private key (or a misbehaving CA).

- Limited prior trust: It is not required to trust other nodes. However, in PoS it is necessary to periodically authenticate the chain state out-of-band to prevent some attacks.

- Simplified management: since CAs are not required, their management overhead is avoided.

- Auditable: allocations and delegations can be tracked back in the blockchain to determine if they originate from the legitimate holder.

Drawbacks:

- PoS does not rely on strong cryptographic guarantees: As opposed to PKI-based systems that rely on strong and well-established cryptographic mechanisms, PoS-based infraestructures ultimately rely on the good behaviour of the high-stakers.

- Slow updates: New transactions have to be verified and added to a block, which adds a delay until nodes recognize them as correct.

- Costly bootstrapping: When a node is activated it has to download and verify the entire blockchain.

- Large storage required: The blockchain grows forever as more blocks are added, blocks cannot be removed.

5.2. Security evaluation

5.2.1. Attacks against a PoS-based consensus algorithm

This section presents a list of the most relevant attacks against a Proof of Stake algorithm and how to mitigate them.

5.2.1.1. Stake grinding

Stake grinding refers to the manipulation of the consensus algorithm in order to progressively obtain more stake, with the goal of signing blocks more frequently with the ultimate goal of taking control of the blockchain. It proceeds as follows: when the attacker has to sign a block, it computes all the possible blocks (varying the data inside them) to find a combination that gives the highest possibility of signing another block in the future. It then signs this block and sends it to the network. This procedure is repeated for all the next signing opportunities. Over time, the attacker will sign more and more blocks until the consensus algorithm will always select the attacker to sign all blocks, thereby having taken control of the blockchain.

To prevent this attack, the source of randomness used to select the signers has to be hard to alter or to predict.

5.2.1.2. Nothing at stake

Nothing at stake is one of the fundamental drawbacks of Proof of Stake and requires careful design based on the incentives of the participants. In common PoS designs, the signers of the new block receive an economical incentive (e.g., Ethereum). However this does not hold in the IP address scenario, since participants should not receive any incentive. The incentive is, as stated before, achieving a consistent view of the IP address space and having a secure Internet.

5.2.1.3. Range attacks

A range attack is performed by creating a fork some blocks back from the tip of the chain. It is conceptually similar to the attack named as 'Risk of overtaking' in Section 3.3.1. In this scenario, the attacker has privately fabricated a chain which (according to the consensus algorithm rules) will be selected over the original one. Benefits of this attack include gaining more stake on the blockchain (this attack could be part of a stake grinding attack) or rewriting the transaction history to erase a payment made in the original blockchain.

The simplest solution to this attack is adding a revert limit to the blockchain, forbidding forks going back more than N blocks. This

provides a means to solidify the blockchain. However, nodes that have been offline for more than N blocks will need an external source that indicates the correct chain. It has been proposed to do this out of band. This is why a PoS blockchain is not purely trustless and requires a small amount of trust.

5.2.2. Attacks against the P2P network

5.2.2.1. DDOS attacks

TBC

5.2.2.2. Transaction flooding

TBC

5.2.2.3. Routing attacks

TBC

6. Other Considerations

6.1. Revocation

TBC

6.2. Key rollover

TBC

6.3. Incentives

TBC

6.4. Storage management

TBC

6.5. Transaction censorship

TBC

6.6. Configuration parameters

TBC

7. Security Considerations

This document aims to understand the security implications of using the blockchain technology to secure IP addresses allocation.

8. IANA Considerations

This memo includes no request to IANA.

9. Acknowledgements

TBD

10. Informative References

[Bitcoin] Nakamoto, S., "Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf", 2008.

[Blockstack]
Ali, et al., M., "Blockstack : A Global Naming and Storage System Secured by Blockchains, USENIX Annual Technical Conference", 2016.

[Ethereum]
The Ethereum project, "https://www.ethereum.org/", 2016.

[Hari2016]
Hari, A. and T. Lakshman, "The Internet Blockchain: A Distributed, Tamper-Resistant Transaction Framework for the Internet. Fifteenth ACM Workshop on Hot Topics in Networks", 2016.

[miningfarm]
Inside a mining farm, "http://www.bbc.com/future/story/20160504-we-looked-inside-a-secret-chinese-bitcoin-mine", 2016.

[Namecoin]
Namecoin, "https://namecoin.org/", 2011.

[Peercoin]
The Peercoin cryptocurrency, "https://peercoin.net/", 2016.

Authors' Addresses

Jordi Paillisse
UPC-BarcelonaTech
c/ Jordi Girona 1-3
Barcelona, Catalonia 08034
Spain

Email: jordip@ac.upc.edu

Alberto Rodriguez-Natal
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: natal@cisco.com

Vina Ermagan
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: vermagan@cisco.com

Fabio Maino
Cisco Systems
170 Tasman Drive
San Jose, CA
USA

Email: fmaino@cisco.com

Albert Cabellos
UPC-BarcelonaTech
c/ Jordi Girona 1-3
Barcelona, Catalonia 08034
Spain

Email: acabello@ac.upc.edu