Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 30, 2018

J. Paillisse
UPC-BarcelonaTech
A. Rodriguez-Natal
V. Ermagan
F. Maino
Cisco Systems
L. Vegoda
Individual
A. Cabellos
UPC-BarcelonaTech
June 28, 2018

An analysis of the applicability of blockchain to secure IP addresses allocation, delegation and bindings. draft-paillisse-sidrops-blockchain-02

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 are 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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 30, 2018.

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 in a distributed way. Formally, it is regarded as a new solution to the Byzantine Generals problem, well-known in fault-tolerant distributed systems [Byzantine].

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 naming system, providing a similar functionality to that of DNSSEC.

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 temporarily store 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.

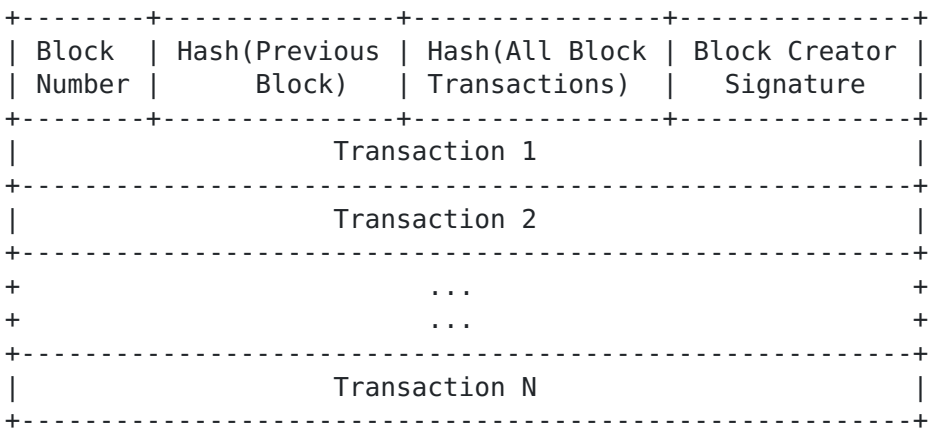


Figure 1.- Common structure of a block

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.

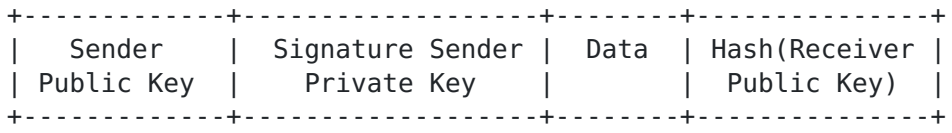


Figure 2.- Common transaction structure in a blockchain

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 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, requiring some computational effort. This is commonly known 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 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 takeover: 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. On the other hand, several ASIC-resistant algorithms are in use in various cryptocurrencies. This is usually achieved with memory-intensive calculations or

frequently changing the mining algorithm. Although they appear to be a promising alternative, vendors react by developing a silicon implementation of the algorithm. In this situation, the developers usually change the algorithm by means of a hard fork [monero]. Ultimately, this becomes an arms-race.

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 pseudo-randomly selected 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:

- It does not require special hardware nor computationally or energy-expensive calculations.
- 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 entities 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 entities that hold large blocks of the 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. Overview of the architecture

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).

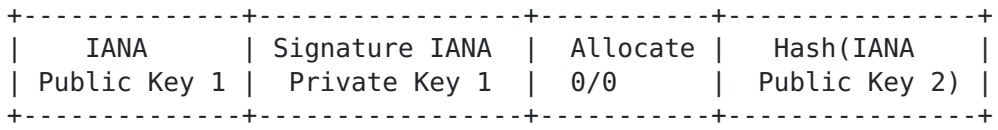


Figure 3.- Genesis transaction

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).

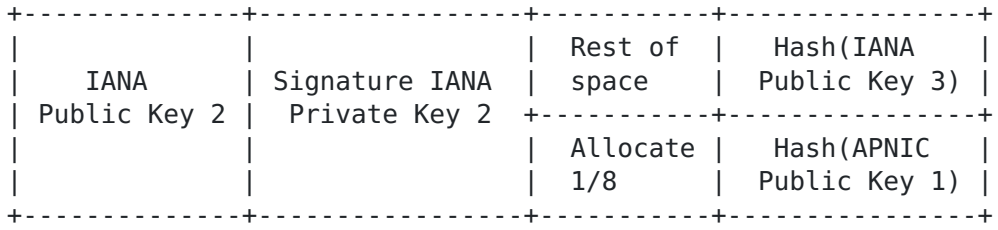


Figure 4.- Example allocation transaction

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.

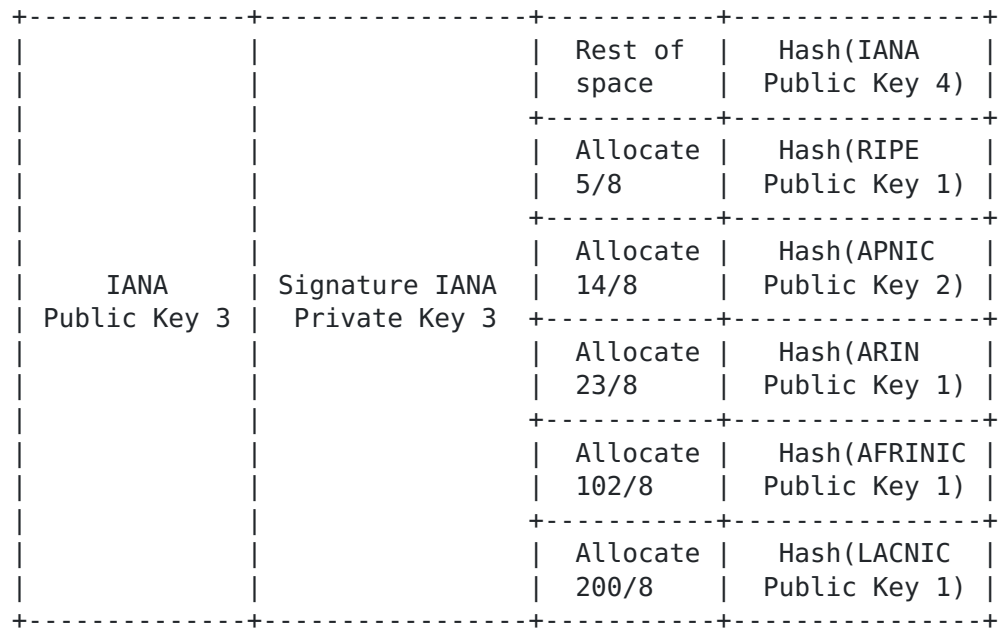


Figure 5.- Example multi-output allocation transaction

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.

Figure 6.- Example delegation transaction

Performing a key rollover is simple, because each transaction has its associated public key, and only depends on the previous transaction. In other words, rekeying means changing the public key only in the holder's transaction. This can be done adding a new transaction with the same data but transferring it to a new public key also controlled by the initial holder (figure 7). This approach lets each entity decide its rekeying policies independently.

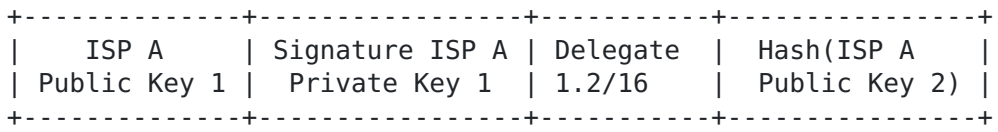


Figure 7.- Example key rollover of a prefix delegation

It is worth noting that this chain can define as many operations as required, for instance storing the binding of AS numbers to the IP prefixes they announce (figure 8).

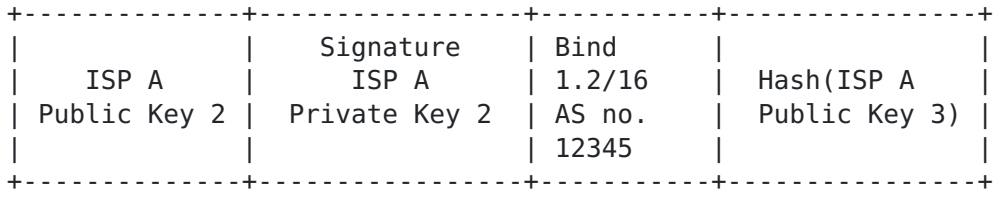


Figure 8.- Example binding of AS number to prefix

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, restrict permissions for customer sub-delegations, etc.

5.1. Support for IPv6 and AS numbers

The allocation and delegation of IPv6 addresses and AS numbers is equivalent to that of IPv4, maintaining the IANA -> RIR -> LIR hierarchy. For example, for IPv6:

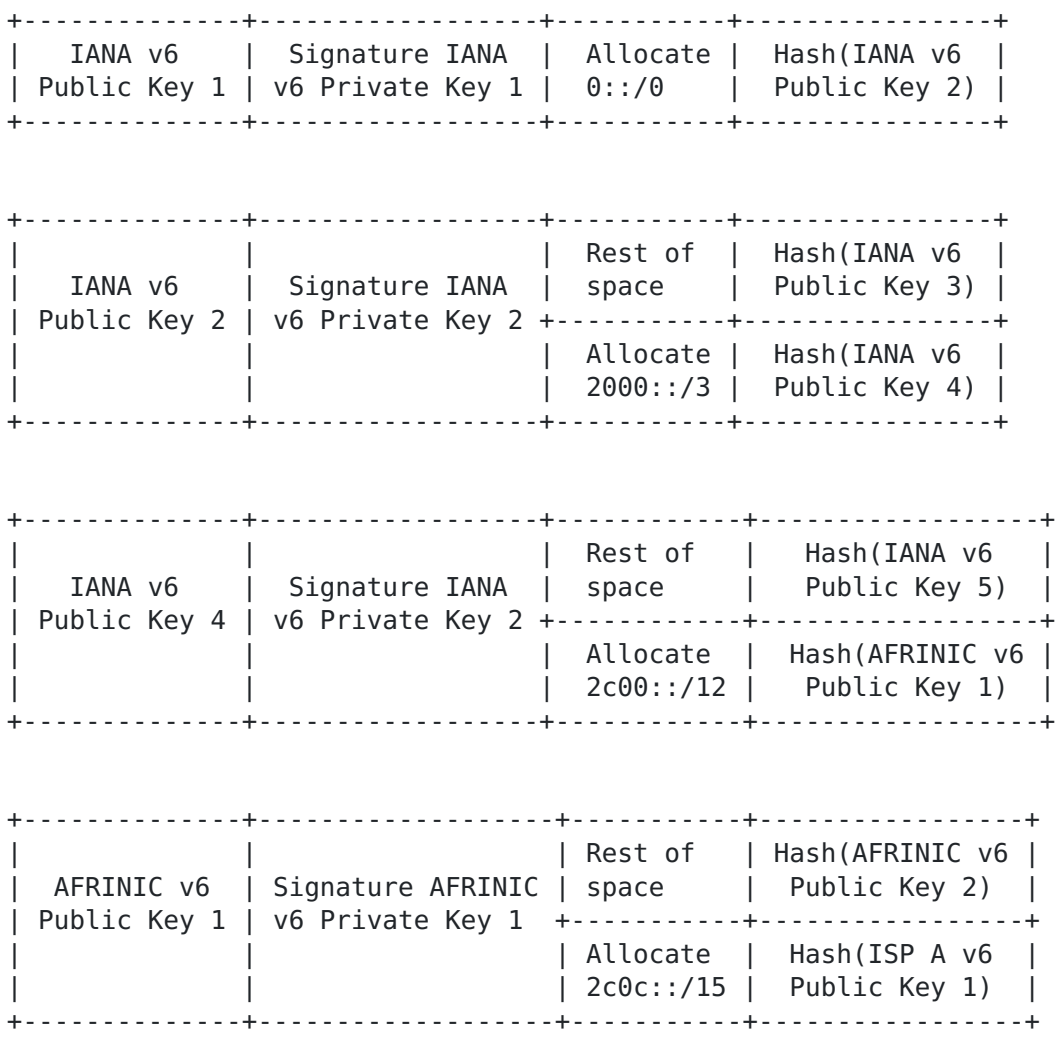


Figure 9.- IPv6 allocation transactions. From top to bottom: genesis transaction, global unicast allocation, AFRINIC allocation and LIR allocation.

The process is equivalent for AS numbers. Besides, in the context of a multi-signature scheme, it is also possible to ask the holder of the AS number to confirm the binding of its AS number to a particular prefix.

5.2. Pros and cons

In this section we analyze the pros and cons of this architecture 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.

- Limited legal liability: since users control their private keys, Internet Registries cannot be held legally responsible of their loss. In turn, this can foster the creation of a unified registry instead of the current five. Ultimately, this would ease cross-registry resource transfers.

- No single point of failure: again, due to the fact that each user controls its private key, the compromise of a user's key does not compromise the entire system. This starkly contrasts with the compromise of a CA, which can potentially invalidate all downstream certificates.

- Simplified state update: PKIs need specific subsystems to update its state (e.g. issue/revoke certificates). On the other hand, in a blockchain all these operations are embedded in it thanks to its transactional nature.

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.

- 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.

5.3. Security evaluation

5.3.1. 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.

5.3.1.1. 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.

5.3.1.2. 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.

5.3.1.3. 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 takeover' 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.

5.3.1.4. Monopolies

A monopoly refers to a single party controlling enough IP addresses so it can sign a significant proportion of new blocks, thus being able to decide which information is written in the chain (e.g., a 51 % attack in Bitcoin). However and in this use-case, this is of less concern since parties do not have a clear incentive to alter normal chain operation. In order to successfully launch this attack a party should control more than 50% of the IP blocks, while this is difficult to achieve and participants do not have a clear incentive to sell/give away blocks of IPs, the attacker would also impact its own infrastructure, making the Internet less secure. Section 7.4 contains more details regarding monopolies.

5.3.1.5. Lack of participation

Participants in a PoS algorithm will not always sign a block, since they might be offline when they are selected or lack incentives. Because of this, the final fraction of high-stakers that sign blocks can be very different from the full set of high-stakers. The direct consequence of this situation is that the portion of participants that decide what goes into the blockchain can be a small set of nodes. If this participation is low enough, it can leave the control of the blockchain to a small amount of people/oligarchy, thus rising security concerns.

5.3.2. Attacks against the P2P network

This section presents attacks directed towards the underlying P2P network used to exchange information among the participants of the blockchain.

5.3.2.1. DDOS attacks

Since blockchains are inherently based on P2P architectures, they present a higher degree of resistance to DDOS attacks than centralized server architectures, provided that the network has a significant number of participants. In addition, it is always possible to keep an offline copy of the blockchain.

5.3.2.2. Transaction flooding

A special type of DDOS attack consists in creating a large amount of legit transactions that transfer a small amount of tokens (i.e. delegate a lot of small IP prefixes). If the number of transactions is large enough, the addition of new transactions can be significantly delayed because not all of them fit into a single block. The effectiveness of the attack also depends on the throughput of the blockchain (transactions/second). Simple solutions may be to limit the granularity upon which IP addresses can be split. Of course, only the legitimate holder of a large amount of IP address can perform this attack.

5.3.2.3. Routing attacks

The underlying P2P network in blockchains does not typically use any security mechanism, e.g. node authentication or integrity of network protocol messages. This enables potentially disruptive attacks. For example, specially located rogue nodes could drop new transactions, which would block updates on the blockchain and leave legit nodes uncommunicated. The effectiveness of this kind of attacks depends on how the P2P algorithm selects peers and the topology of the P2P network.

However, the most potentially dangerous attack of this type are network partitions, i.e. isolating a group of nodes from the rest of the network so they cannot communicate each other (e.g., [Apostolaki2017]). The consequence of this attack is that two versions of the blockchain are created, one at each network partition. When the partition disappears and the nodes reconnect one of the two chains will be discarded, causing a service disruption. It is worth noting that Bitcoin has suffered similar attacks [realrouteattack].

5.3.2.4. Transaction censorship

When a node adds a block it chooses arbitrarily which transactions are added into it, i.e. no specific rules control how transactions are added to a block. This enables a node to selectively add some transactions and intentionally exclude others, with the consequence that some transactions may be never added to the blockchain. In the context of IP addresses, this may be performed by a competing ISP to prevent another ISP from executing a certain modification. Possible solutions revolve around:

- Giving more priority to older transactions (similarly to Bitcoin).
- Punishing nodes that exhibit this kind of behaviour, e.g. removing part of their block of IP addresses or lowering their chance of adding blocks.

6. Revocation

Due to the irreversible nature of transactions, once a block of IP addresses has been allocated to an entity it is not possible to modify or remove it, as opposed to CRLs (Certificate Revocation Lists). However, due to operational issues (compromised or lost keys, human mistake, holder misbehaviour, etc) it is critical to provide a way to recover a block of addresses. Moreover, since IP addresses are a finite public good they cannot be lost. Taking into account that a blockchain can enforce any rules its participants agree upon, this section presents some possible approaches to implement revocation, such approaches should not be considered as mutually exclusive. The revocation procedure must be discussed among the community to achieve consensus between the relevant players (IANA, RIRs, ISPs, institutions, etc).

All these mechanisms present different balances of power between the current holder and the entity whose asset is being revoked. Behind all these mechanisms there is a fundamental tradeoff between trusting an upstream provider of the addresses and retaining full control of the block of addresses.

Regardless of the revocation policy and as opposed to traditional PKI systems, each IP prefix delegation only depends on the private key of the holder of such IP block. As such, it does not need to trust a CA or a chain of certificates. Only by means of this private key the IP delegation can be altered.

6.1. Expiration time

A simple approach to allow revocation is adding a lease time (i.e, time-to-live) to the blocks of addresses. After the lease ends, the new holder of the address block automatically becomes the previous one, or addresses are transferred to a default holder. As stated before, this revocation procedure should be enforced by the rules of the blockchain, this means that participants would not recognise expired allocations as valid.

6.2. Multi-signature transactions

A multi-signature transaction is a transaction with more than one associated public key. In other words, a transaction is considered valid if it has, for instance, 2 out of 5 valid signatures. This way, 3 keys can be lost but with the renaming 2 keys the block of addresses can be recovered. This approach exemplifies the aforementioned tradeoff in trust, since the holder of the block of addresses must trust the owners of the keys participating in the multi-signature transaction. For instance, if some of these keys are owned by IANA or an Internet Registry, we can return part of the control over the allocation to them.

6.3. Revocation transaction

A simpler approach than multi-signature transactions is creating a 'revocation' transaction. When a block of address is required to be reassigned without the consent of the current holder, a revocation transaction (specifying the new holder) is inserted in the blockchain. This transaction should be issued either by a consensuated authority or by a disputing entity. The revocation transaction should be resolved by either accepting the revocation transaction automatically when issued by the accepted authority or by means of out-of-band mechanisms when issued by a disputing party.

6.4. Heartbeat transaction

Another approach involves issuing a heartbeat transaction every N days, signalling to the network that the holder still owns the key associated with that particular resource. If the holder fails to issue this transaction, the blockchain considers that the resource is automatically returned to the registry.

6.5. Out-of-band mechanisms

Disputes regarding transactions can be resolved by means of out-of-band mechanisms, e.g, discussion, court, etc. In order to reflect the decision of this out-of-band mechanism the blockchain must be modified. Since this represents a deviation from the rules, it must be done through a hard blockchain fork. Although cumbersome and complex, this is feasible from a technical standpoint.

6.6. A simple revocation protocol

Here we present a simple revocation protocol to handle accidental key loss:

1. On detecting the key loss, the holder notifies the registry, e.g. via e-mail.

2. The registry issues a revocation transaction, similar to the one in section Section 6.3.

3. The current holder has a fixed period of time to issue a transaction re-claiming the resource. This transaction must be signed by the private key associated with the claimed resource.

4. If the holder issues the transaction, it retains the resource.

5. Otherwise, after the fixed time interval, the blockchain considers that the resource is returned to the registry, so it can be re-allocated.

This protocol combines two of the aforementioned techniques, and allows to balance power between resource holders and registries. Registries can revoke lost or unclaimed resources, while address holders can retain them if the registry misbehaves or its key is compromised. However, it should be noted that this protocol does not protect from stolen keys.

The time interval can be different depending on the nature of the revocating entity. For example, IANA -> RIR allocations could wait a couple of weeks, whereas RIR -> LIR allocations could go faster with a 72 hours notification period.

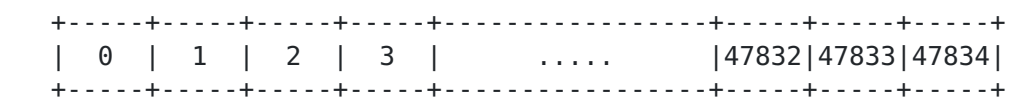
7. Other Considerations

7.1. Storage management

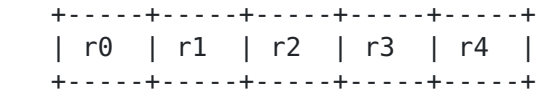
The never ending size of the blockchain presents a potential scalability issue. At the time of this writing, mature blockchains

like Bitcoin require more than 100 GB of storage. Simply deleting or summarizing old transactions degrades the security of PoW-based chains, since their security relies on the computing power required to generate them. The longer they are, the harder they are to attack.

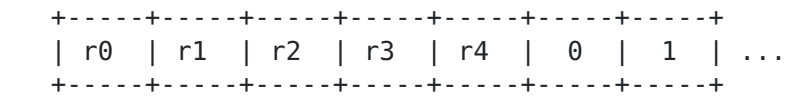
However, PoS-based chains do not rely on computing power and hence, space-saving strategies do not degrade the security. For instance a simple solution could be to, once the PoS-based chain reaches a certain storage size, summarize a subset of the older transactions. In what follows we overview this strategy:



9.1 Old blockchain



9.2 Write present state to special reset blocks



9.3 Continue working after the reset blocks

Figure 10.- A simple technique to reduce blockchain storage

This approach reduces bootstrapping cost, it is worth noting that this strategy requires trust on the reset blocks, such blocks can be obtained with an out-of-band mechanism (see Section 5.3.1.3 for further information).

7.2. Proof of Networking?

In this section we speculate how one could design an equivalent of Proof-of-Work (PoW) for networks. Conceptually, PoW is a proof of computational resources, can we devise a proof of networking resources? It could be thought that a PoW equivalent may exist in the context of networks, i.e., an equivalent to spending computer cycles in a network. Some resources unique to networks are bandwidth, computation of checksums, number of BGP peers, etc. Hence, we could envisage a blockchain secured by the resources inherent to its participating computer networks. As long as half of the resources were controlled by honest members, security is guaranteed. For example, bandwidth could be a potential candidate; however it does not satisfy two key features present in PoW:

- Asymmetry: the proof has to be hard to generate but fast to verify.

- Verifiability: it has to be possible to embed the proof in the block in order to account for the spending of resources.

In this context, Proof-of-Networking is an open research issue .

7.3. Configuration parameters

Configuration parameters refer to a set of values:

- Block creation rate

- Maximum block size

- Other parameters related to the consensus algorithm

These parameters, beyond regulating the operation of the blockchain also have an influence on its performance. For example, a small block size increases propagation speed (thus consensus can be reached faster) but reduces the number of transactions per second that the blockchain can handle. As an example, in Bitcoin, the 10-minute block creation rate seeks to balance fast confirmation times and reduced probability of forks [Antonopoulos2015]. Experimental deployments and operational requirements should help tuning such parameters.

7.4. PoS algorithm design particularities

The particular use case of IP addresses presents some characteristics that the PoS algorithm should take into account:

Monopoly prevention: As described in section Section 5.3.1.4, monopolies can pose a threat to a PoS blockchain. In order to

prevent a small number of large-stakers from controlling the chain, we can design the PoS algorithm to have a smart mapping of IP prefixes to the weight of the random selection. A potential solution could be fine-tuning the weighting of IP addresses (slightly biasing the choice towards medium-sized holders), in order to reduce the power of high stakers. This can provide an upper-bound of the maximum number of addresses that a party can hold to avoid monopolies (ideally as high as possible).

IPv6 support: Large parts of the IPv6 address space remain unallocated and still owned by IANA (at the time of writing this document, less than 0.5% of v6 address space has been allocated to the RIRs). The PoS algorithm should ignore this space (not count it) to avoid IANA signing nearly all v6 blocks and thus, preventing an IANA monopoly.

IPv4/v6 stake isolation: Since there are more IPv6 addresses than IPv4, this creates an imbalance of power in a PoS blockchain: randomly selecting from both pools of addresses naturally causes v6 holders signing more blocks than v4 holders. Thus, some kind of isolation between v4 and v6 stake is required. For example, we could alternatively generate blocks with only v4 or v6 transactions, signed by a v4 or v6 holder, respectively.

7.5. Candidate PoS consensus algorithms

There are several existing PoS algorithms that could satisfy the requirements of a blockchain for IP addresses. The following list presents three of them that have been claimed by the authors to be proven mathematically correct. A substantial difference among them is the supported portion of offline participants. The list does not pretend to be exhaustive.

Algorand: [Algorand] leverages a multi-step protocol to provide a verifiable random selection of the block signer. The most relevant features of Algorand are:

* A cryptographic sortition mechanism to randomly select the participants in each step of the protocol, based on Verifiable Random Functions [I-D.irtf-cfrg-vrf].

* The decoupling of block creation and block signing, to avoid stake grinding attacks.

* A new Byzantine Agreement protocol (BA*) to achieve consensus.

* Player replaceability: Algorand uses a different set of participants for each of its steps. Thus, malicious participants from one step cannot influence the following.

* Upper bound of 1/3 of dishonest players.

Ouroboros: [Ouroboros] presents a similar approach to Algorand: first, it selects a subset of users. Then, these users perform the random selection by means of a secure multiparty computation. As opposed to Algorand, however, this subset lasts for several blocks, called epochs. In addition, Ouroboros assumes that the majority of participants are online when they have to participate in the protocol, and that they remain offline only short periods of time.

Snow White: The [SnowWhite] protocol improves on the previous two by supporting also large amounts of offline participants, as long as the majority of online members are honest. They call this property 'Robustly Reconfigurable Consensus'. Snow White also leverages a random function to decide if a participant has to sign a block, and defines epochs similarly to Ouroboros: after each epoch, participants for the new one are calculated.

7.6. Privacy concerns

In order to protect privacy, the blockchain should not contain Personally Identifiable Information (PII). This is due to the fact that data in the blockchain cannot be removed and that it is a public ledger, accessible by anyone. Instead, PII like contact emails or postal addresses should be stored in the Registry's database (e.g. RIPE Database). Ideally, the blockchain should contain the minimum amount of data for correct operation, that is: public keys, blocks of IP addresses, AS numbers and their bindings (figure 11).

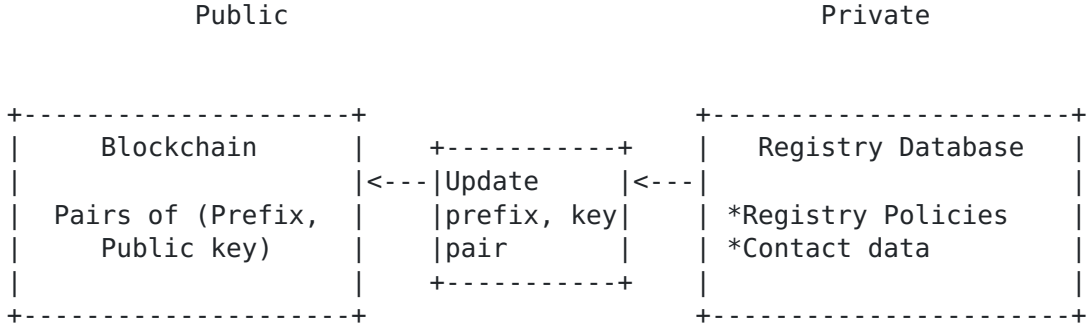


Figure 11.- Data flow between Registry and blockchain

7.7. Governance

Blockchain does not mean anarchy. In fact, any blockchain requires a governing entity that determines its rules and ensures that all of its participants agree on its operation. The need of governance is illustrated by the recent Bitcoin Cash fork [Bitcoincash]. Due to a disagreement among Bitcoin users, they created a Bitcoin hard fork called Bitcoin Cash. This split the Bitcoin blockchain in two, causing some confusion. Proper governance should avoid such situations.

This particular use case is not an exception: all concerned parties (IANA, RIRs, ISPs, etc) should reach consensus regarding which rules are enforced in the blockchain. For example, dispute resolution or revocation procedures.

8. Implementations

There are several implementations to secure the allocation of IP prefixes. They present different scopes and levels of maturity.

- IPchain uses a Proof of Stake algorithm and is specifically tailored for the allocation of IP addresses. Its performance has been evaluated [IPchain], and has been open-sourced [IPchain-repo].

- [BGPCoin] runs on top of the Ethereum blockchain and provides similar features to IPchain. It has not been possible to find open-source code.

- Another project identically named BGPCoin is designed to allow ISPs to exchange peering agreements and route advertisements by means of a blockchain [BGPCoin-repo]. It uses a hybrid PoW/PoS algorithm and has its own cryptocurrency.

9. Security Considerations

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

10. IANA Considerations

This memo includes no request to IANA.

11. Acknowledgements

The authors wish to thank Jordi Herrera-Joancomarti, Andreu Rodriguez-Donaire and Jordi Baylina for their helpful discussions

about Bitcoin and blockchain technology, as well as Marco Chiesa for the heartbeat transaction idea.

12. Informative References

[Algorand]
Gilad, Y., Hemo, R., Micali, S., Vlachos, G., and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies. Proceedings of the 26th Symposium on Operating Systems Principles (pp. 51-68). ACM.", 2017.

[Antonopoulos2015]
Antonopoulos, A. M., "Mastering Bitcoin, available online: http://chimera.labs.oreilly.com/books/1234000001802/index.html", 2015.

[Apostolaki2017]
Apostolaki, M., Zohar, A., and L. Vanbever, "Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. 2017 IEEE Symposium on Security and Privacy (SP). ", 2017.

[BGPCoin-repo]
BGPCoin GitHub repository, , "https://github.com/bgpcoin", 2018.

[BGPCoin] Xing, Q., Wang, B., and X. Wang, "POSTER: BGPCoin: A Trustworthy Blockchain-based Resource Management Solution for BGP Security. ACM Conference on Computer and Communications Security (CCS) 2017", 2017.

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

[Bitcoincash]
Bitcoin split in two, here's what that means, , "http://money.cnn.com/2017/08/01/technology/business/bitcoin-cash-new-currency/index.html", 2017.

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

[Byzantine]
Lamport, L., Shostak, R., and M. Pease, "The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems", 1982.

[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.

[I-D.irtf-cfrg-vrf]
Goldberg, S., Reyzin, L., Papadopoulos, D., and J. Vcelak, "Verifiable Random Functions (VRFs)", draft-irtf-cfrg-vrf-01 (work in progress), March 2018.

[IPchain-repo]
IPchain GitHub repository, , "https://github.com/OpenOverlayRouter/blockchain-mapping-system", 2018.

[IPchain] Paillisse, J., Ferriol, M., Garcia, E., Latif, H., Piris, C., Lopez, A., Kuerbis, B., Rodriguez-Natal, A., Ermagan, V., Maino, Fabio., and A. Cabellos, "IPchain: Securing IP Prefix Allocation and Delegation with Blockchain, arXiv preprint: https://arxiv.org/abs/1805.04439", 2018.

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

[Ouroboros]
Kiayias, A., Russell, A., David, B., and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol. Annual International Cryptology Conference (pp. 357-388). Springer, Cham.", 2017.

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

[SnowWhite]
Bentov, I., Pass, R., and E. Shi, "Snow White: Provably Secure Proofs of Stake. IACR Cryptology ePrint Archive, 2016, 919.", 2016.

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

[monero] Monero PoW algorithm update, , "https://www.ethnews.com/monero-team-mulls-changing-pow-algorithm-to-preempt-asic-miners", 2018.

[realrouteattack] Hacker Redirects Traffic From 19 Internet Providers to Steal Bitcoins, , "https://www.wired.com/2014/08/isp-bitcoin-theft/", 2014.

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

Leo Vegoda
Individual
4712 Admiralty Way, #152
Marina del Rey, CA 90292
USA

Email: leo@vegoda.org

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

Email: acabello@ac.upc.edu