IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 29, 2019, accepted November 10, 2019, date of publication November 18, 2019,
date of current version November 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954096

Decentralized Trust in the Inter-Domain Routing Infrastructure

JORDI PAILLISSE[1], JAN MANRIQUE[2], GUILLEM BONET[3], ALBERTO RODRIGUEZ-NATAL[4], FABIO MAINO[2], AND ALBERT CABELLOS[1]

[1]Computer Architecture Department, UPC-BarcelonaTech, 08034 Barcelona, Spain
[2]Quercus Technologies, 43203 Reus, Spain
[3]Sogeti, 08018 Barcelona, Spain
[4]Cisco Systems, San Jose, CA 95134, USA

Corresponding author: Jordi Paillisse (jordip@ac.upc.edu)

This work was supported in part by the Spanish MINECO and EU FEDER (ALLIANCE Project) under Contract TEC2017-90034-C2-1-R, in part by the Catalan Institution for Research and Advanced Studies (ICREA), and in part by a Cisco Grant.

ABSTRACT Inter-domain routing security is of critical importance to the Internet since it prevents unwanted traffic redirections. The current system is based on a Public Key Infrastructure (PKI), a centralized repository of digital certificates. However, the inherent centralization of such design creates tensions between its participants and hinders its deployment. In addition, some technical drawbacks of PKIs delay widespread adoption. In this paper we present IPchain, a blockchain to store the allocations and delegations of IP addresses. IPchain leverages blockchains' properties to decentralize trust among its participants, with the final goal of providing flexible trust models that adapt better to the ever-changing geopolitical landscape. Moreover, we argue that Proof of Stake is a suitable consensus algorithm for IPchain due to the unique incentive structure of this use-case, and that blockchains offer relevant technical advantages when compared to existing systems, such as simplified management. In order to show its feasibility and suitability, we have implemented and evaluated IPchain's performance and scalability storing around 350k IP prefixes in a 2.5 GB chain.

INDEX TERMS Blockchain, decentralization, inter-domain routing, IP prefixes, IP networks, Proof of Stake.

I. INTRODUCTION

Inter-domain routing security is a pressing issue in today's Internet. In a nutshell, inter-domain routing security encompasses the correct announcement and propagation of IP prefixes across the Autonomous Systems (AS) that conform the Internet. Currently, the protocol that communicates these announcements is BGP (Border Gateway Protocol, RFC 4271). BGP allows ISPs and other Internet companies to announce routes, i.e. how to reach a specific destination. BGP security is typically based on manual and careful configuration via out-of-band mechanisms where network operators communicate each other which prefixes to announce. Hence, an accidental misconfiguration or a malicious attacker controlling a BGP router can disrupt normal Internet routing [1]. This can lead to denial of Internet services, traffic redirection, data leaks, etc.

One of the most relevant attacks to the Inter-domain routing infrastructure is known as prefix hijacking. Since BGP messages are not authenticated, it is easy to perform a BGP hijack by forging BGP announcements and propagating them to neighboring ASes. There is a long history of prefix hijacks on the Internet. As an example of this, in 2008 the Pakistani government ordered national ISPs to censor YouTube. Due to a configuration error, they attracted large portions of non-Pakistani YouTube traffic which resulted in the service being down during 2 hours worldwide [2].

Given the severity of these attacks, the IETF (Internet Engineering Task Force) has designed a solution to Inter-domain routing security by means of the RPKI (Resource Public Key Infrastructure, RFC 6480), a PKI repository to record the legitimate owners of IP prefixes, AS numbers and ROAs (Route Origin Authorization, a certificate to allow an AS to announce an IP prefix). Despite these efforts, RPKI deployment is slower than expected: only ∼14% of the total /24 IPv4 address blocks owned by the five Internet Registries are protected by the RPKI (fig. 1).

There are several reasons that contribute to this limited deployment, related to technical issues but also to policy aspects. By policy aspects we refer to RPKI's

The associate editor coordinating the review of this manuscript and approving it for publication was Noor Zaman.

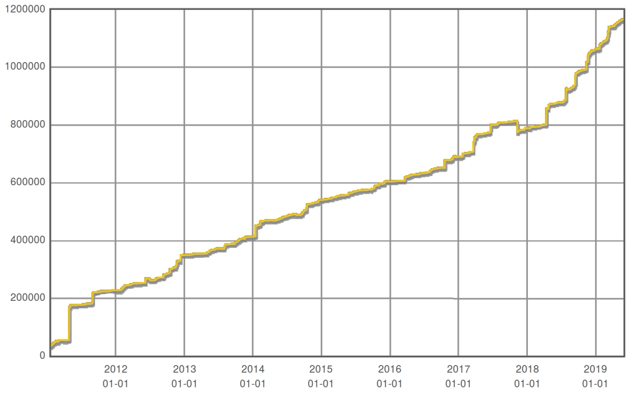


FIGURE 1. Amount of RIPE's IPv4 address space covered by ROAs, in /24 units. Of the five Registries, RIPE is the one with highest RPKI adoption [3].

inherent centralization, that leaves ultimate control of Internet routing to the RPKI Certification Authorities (CAs), in this case the five Regional Internet Registries (RIRs) [4]. This centralized security model does not align well with the current situation of the Internet, which is at a crossroads with different competing visions on how it should be [5], and risks splintering into isolated networks known as *splinternet* [6]. In other words, we face a dilemma between centralization and decentralization.

On the technical side, the RPKI presents various obstacles to widespread deployment: management complexity (PKIs are cumbersome to manage, e.g. when performing a key refresh), implementation challenges [7], and concerns on transparency [8]. In addition, deploying these extensions is not trivial and requires trained staff [9] and financial investment.

In the light of this situation, we propose securing Inter-domain routing by storing IP address allocation data in a blockchain. Thanks to its decentralized nature, we can distribute trust among all of its participants (i.e. all the owners of IP addresses). This way, each entity maintains its independence but at the same time they have a common framework to agree on routing security, thereby reducing the incentives to isolate parts of it. In addition, we can create flexible trust models that capture the complexities of geopolitics and replace single points of control (CAs) with global agreement (blockchain consensus algorithms).

Moreover, with a blockchain we can address the aforementioned technical drawbacks: (i) simplify management, especially regarding common PKI operations such as key rollover, (ii) offer auditability: blockchain's append-only ledger can detect possible configuration errors even before a modification [10], and (iii) create a consistent vision of the state that does not depend on additional systems (i.e. PKI Certificate Revocation Lists, CRLs), because all the required operations can be embedded in the blockchain.

In this paper we present IPchain, a blockchain to store IP address allocation and delegation data. The underlying argument is that IP addresses are very similar to crypto-coins, e.g. they are unique or can be divided into smaller amounts. Just like in Bitcoin users send money, participants in IPchain can transfer IP addresses. For example, an ISP can store its IP prefixes in the chain, along with the originating AS number. Then, other ISPs can use this data to verify the origin of BGP messages associated with these prefixes.

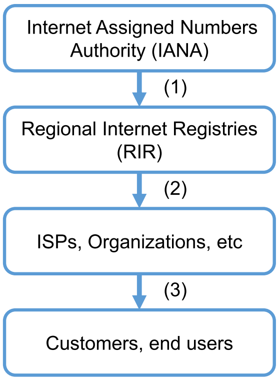

Internet Assigned Numbers Authority (IANA)
(1)
Regional Internet Registries (RIR)
(2)
ISPs, Organizations, etc
(3)
Customers, end users

FIGURE 2. IP address allocation hierarchy.

We have developed a prototype to assess the feasibility of our proposal, focusing on scalability and performance. Such prototype follows the blockchain transaction paradigm to allocate and delegate IP prefixes. We leverage a Proof of Stake consensus algorithm, i.e. select randomly the signer of the next block among all participants, weighted by their number of IP addresses. Finally, we performed an experimental evaluation: we converted the IP prefixes of the Internet Registries into blockchain transactions and stored approximately 70% of them in a chain of 2.5 GB.

This paper is an extension of an earlier version that was presented in a conference [11]. Specifically, we have introduced a new design for the Proof of Stake algorithm (sec. VI-A), new evaluation metrics for such algorithm (sec. VII), an analysis of the feasibility of a PoS monopoly in real life (sec. IV-C), and a long-term storage estimation (sec. VII-C).

II. BACKGROUND: IP ADDRESS ALLOCATION AND RPKI ARCHITECTURE

The allocation of IP addresses follows a hierarchical scheme. It is usually formed of three tiers (fig. 2): the Internet Assigned Numbers Authority (IANA), the Regional Internet Registries (RIRs) and ISPs. Initially, IANA holds all the address space, since it is charge of Internet numbers. Then, it transfers large blocks of addresses to the RIRs (1). Afterwards, the Registries delegate smaller blocks to their customers, usually ISPs (2). Finally, ISPs can delegate blocks to their customers (3). The procedure is equivalent for AS numbers.

In the RPKI we find the same hierarchical structure with digital certificates, which authenticate the allocation of IP prefixes and AS numbers (fig. 3). In summary, there are two types of certificates: Resource Certificates (RC) and Route Origin Authorizations (ROA). RCs bind IP prefixes or AS numbers to a public key (1), and ROAs specify which AS

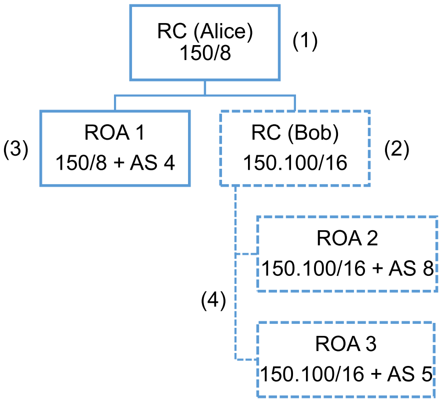


FIGURE 3. Sample RPKI certificate hierarchy.

number can advertise a particular IP prefix (3). It is also possible to sub-allocate resources to other entities (2) or issue more than one ROA (4).

This way, network operators download the certificates and use them to verify BGP announcements. If the (IP prefix, AS number) pair in the BGP message does not match the corresponding certificate in the RPKI, the announcement is considered invalid.

In this architecture, trust is centralized in the five RIRs, who act as CAs and own the Trust Anchor used to validate all downstream certificates.

III. WHY BLOCKCHAIN?

In this section we discuss the advantages provided by a blockchain when compared to the RPKI. We focus both on technical and policy benefits. By policy we refer to the possibility of modifying current trust architectures in order to better align the interests of all parties.

A. TECHNICAL ADVANTAGES

Consistent vision of the state: Both in Bitcoin and the RPKI, we need a mechanism to update the state and notify all participants, so that all of them agree on the same data. While the RPKI makes use of specific protocols to update state (e.g. RFC 8181 or Certificate Revocation Lists), in a blockchain this can be directly encoded in its transactions.

Auditability: Since blockchain transactions cannot be eliminated, we can easily detect updates, for example, when a new ROA is issued for the same IP address. Moreover, this persistence prevents unwanted deletions that could affect other users [10]. Even though we can build auditability systems for PKIs[1], a blockchain has this feature out of the box.

Simplified management: common RPKI operations, such as key refresh or certificate revocation are complex and require multiple steps or dedicated subsystems. On the other hand, a blockchain makes those operations much more simple, usually it is only necessary to add a new transaction. In a PKI, when we perform a key rollover we have to re-sign all downstream certificates[2], and if we want to revoke certificates we need manifests and CRLs. As we said, in a blockchain we can perform these two operations with a new transaction.

Privacy: Blockchain transactions are not linked to the user's identity, just to a public key. It is worth noting that the RPKI also offers privacy, because its certificates do not contain identity information.

B. POLICY ADVANTAGES

The main benefit of using blockchain in our scenario is the decentralization of trust. As mentioned earlier, RPKI users (typically ISPs) have to trust the five RPKI CAs (one for each RIR), which act as central points of trust, and can arbitrarily revoke any downstream certificate [4]. This situation can be uncomfortable for ISPs, because IP prefixes are a key asset for most of them (their connectivity is based on proper attribution of IP addresses).

On the contrary, with a blockchain we can tackle these concerns, because their inherent decentralization leaves control of resources to the owner of the public-private key pair. Thus, in a blockchain we can effectively shift the power from CAs to the users, so that after the prefix allocation, users do not depend anymore on the actions of the CA.

Moreover, due to the fact that we can enforce complex policies inside a blockchain, it is possible to define flexible trust schemes between RIRs and users (c.f. section V-E), i.e. a middle ground between complete centralization and decentralization.

IV. WHICH CONSENSUS ALGORITHM?

Consensus algorithms are probably the most important building block of a blockchain. This section details the motivations when deciding which one to use for the use-case of securing IP address allocation and delegation.

A. PROOF OF WORK

In Proof of Work (PoW), nodes in the blockchain have to solve a complex mathematical problem to add a block, thus requiring some computational effort. The definitive chain is the one with most computing power spent to create.

Despite its widespread usage, PoW is not suitable for our use case. The main reason is that the security of a PoW chain is directly linked to computing power. In other words, if we can accumulate enough computing power, we can rewrite the blockchain with false data (e.g., incorrect delegations of IP addresses). This is very expensive in blockchains accounting for millions of participants (like in Bitcoin or Ethereum), due to the large amount of computing power powering them. However, in our situation this kind of attack is feasible: consider that the current number of Autonomous Systems in the Internet is only around 65k [12].

[1] https://datatracker.ietf.org/wg/trans/about/

[2] RFC 6489 is specifically devoted to key rollover in the RPKI.

B. PROOF OF STAKE FOR IP PREFIX ALLOCATION AND DELEGATION

In a Proof of Stake (PoS [13]) blockchain, participants with more assets/coins are more likely to add blocks. Like in PoW, the algorithm randomly selects one participant to add a block, but it takes into account emphhow many coins each user has. The underlying idea is that users with more coins (stake) have an incentive to contribute in the chain because they are its primary users.

Taking into consideration these particularities, we advocate that PoS is the most suitable option for our use-case, due to three key reasons.

First, in PoS only blockchain users can make modifications, i.e. we don't depend on external actors and their computing power. This is of paramount importance in our scenario, because users that own a large quantity of IP address will have higher chance to add blocks. Usually, such participants also profit from an Internet that operates properly, so they have a clear motivation to keep its normal operation. In other words, blockchain users do not have any incentive to forge information because they would suffer the consequences: an insecure Internet [11].

Second, with a PoS algorithm we can reduce the risk of takeover, i.e. buying a large amount of assets in order to accumulate enough stake to rewrite the blockchain. In our context, this means buying a large amount of IP prefixes from other participants. However, it is not clear that an attacker may be able to perform such attack, because the other users lack a clear reason to sell their IP addresses: as we mentioned previously, they are an important economical asset for most ISPs.

And third, with a PoS algorithm we benefit from a low computational cost and we don't need special hardware. These two facts lower the entry barrier for new participants in the blockchain.

C. POS RESISTANCE TO MONOPOLIES

Nevertheless, PoS presents a fundamental weakness: monopolies. If a participant controls half or more of the assets, it will eventually take control of the blockchain. In order to determine if this could happen in a chain for IP addresses, we have calculated how many addresses own the five RIRs and selected countries or political unions. Fig. 4 presents the percentage of IPv4 addresses of each, derived from IP to AS mappings[3] and CAIDA AS to organization mappings. As the figure shows, no country or RIR owns more than 40% of addresses, rendering PoS resistant to monopolies in this scenario (we are assuming that a collusion of two or more of the presented entities is highly unlikely). A similar analysis focusing on individual companies reveals that the one with most prefixes holds 3.5% of all the advertised IPv4 addresses.

Furthermore, in some PoS algorithms we can configure the minimum number of participants needed to create a monopoly, e.g. a participant needs to accumulate at least 75% of the total stake in order to successfully perform an attack. This way, we can adapt to the changing political situation of the Internet, increasing this threshold if required (sec. VI-A, *PoS Consensus Algorithm*).

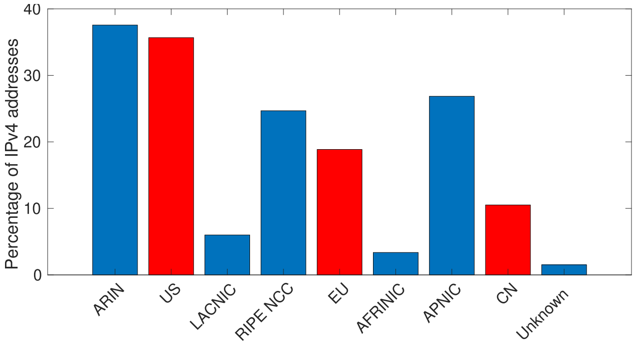


FIGURE 4. Percentage of IPv4 addresses of each RIR. Selected countries/unions are also included in the corresponding RIR.

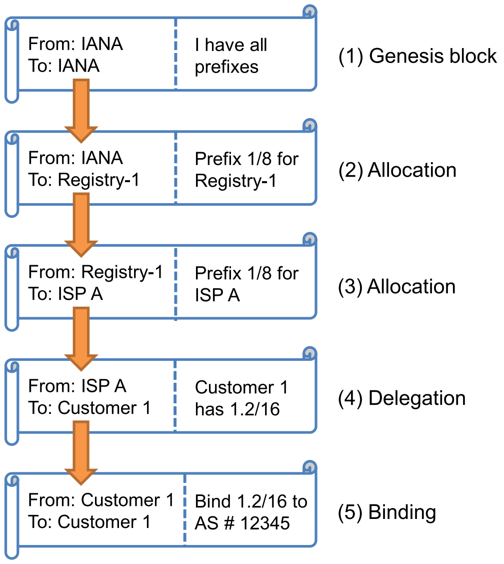


FIGURE 5. Transaction workflow example.

V. ARCHITECTURE OF IPCHAIN

In this section we describe the architecture of IPchain regarding workflow, intended deployment, PoS consensus algorithm, and solutions to recover lost keys.

A. IP PREFIXES AS COINS

IP prefixes share some fundamental characteristics with the coins or assets we find in any blockchain:

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

Taking into account these similarities, we can devise a blockchain to record and transfer IP prefixes, equivalently to a financial blockchain. By chaining different transactions we can replicate the allocation hierarchy of the RPKI in a blockchain and create a consistent registry of owners of IP prefixes. Figure 5 shows the intended deployment of such

[3] https://iptoasn.com/

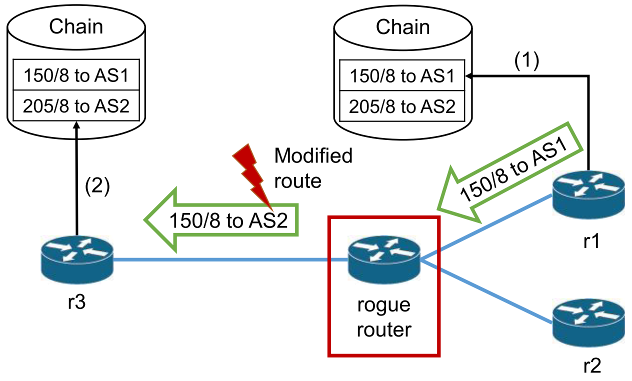


FIGURE 6. Sample usage scenario of IPchain.

blockchain: first, IANA, as the top-level regulator of Internet numbers writes transactions assigning all the address space to itself (1). Ideally, this first transaction is encoded in the genesis block. Second, IANA transfers large blocks of addresses to the RIRs (2). Third, the Registries allocate prefixes to ISPs (3), which in turn delegate them to their customers (4). Finally, customers bind metadata to their prefixes, such as their AS number (5).

The genesis block contains all existing prefixes, so any participant can download the blockchain, validate all the transactions and determine the legitimate owner of a particular prefix.

B. OVERVIEW

Fig. 6 presents a sample of IPchain's workflow. First, router r1 writes in the chain its legitimate prefix and associated AS number (1). Now, consider that the announcement propagates through the network and is modified by the rough router (center). When router r3 receives the announcement of `150/8 to AS2`, it can check in the blockchain (2) if 150/8 should be originated by AS2. In this case 150/8 should be originated by AS1, so the announcement is deemed invalid.

It is worth noting that all the blockchain processes occur offline (equivalently to the RPKI) in a standard server, so they don't need to be co-located with the router. Usually, the server generates a list of valid (IP prefix, AS number) and sends it to the BGP routers.

C. PoS CONSENSUS ALGORITHM

As mentioned in section IV-B, PoS fits the requirements of our use-case. In this scenario, the selection of the next block signer works in the following way:

1) Count the number of addresses of each participant
2) Generate a random number
3) Select one of the participants with the random number weighted by their number of addresses

We must remark that the random number is generated in a distributed fashion, i.e. participants exchange several messages to create the same random number. This way, they will also select the same signer. The key challenge in a PoS algorithm is creating a publicly verifiable random number in a distributed environment. We can find examples in the literature on how to this, such as the Round-Robin Random Number Generator [14] or the Shamir Secret Sharing scheme [15]. Recently, new algorithms have been proposed specifically designed for blockchains (sec. VIII), such as Snow White [16] or Ouroboros [17].

Usually, PoW and PoS algorithms include some kind of reward for new blocks, like Bitcoin, or a punishment for (e.g. Ethereum Casper). In our scenario, we consider both unnecessary: first, the reward is the security of the routing infrastructure; apart from the fact that a PoS algorithm does not require a significant financial investment. Second, in our case a punishment would mean removing the IP addresses of a participant, however, this is very unlikely in real life and can have harsh consequences.

D. SUPPORTED OPERATIONS

We define the following operations for IPchain, that are equivalent to those in the PRKI:

Allocate: Assign a block of IP prefixes to an entity, allowing it to further allocate or delegate it to other entities.

Delegate: Like *Allocate*, but without the permission to further allocate prefixes to other entities.

Metadata: Add additional data to a prefix, e.g. AS number authorized to announce the prefix.

E. FLEXIBLE TRUST: REVOCATION

Blockchain transactions are irreversible, i.e. once we have allocated an IP prefix to an entity, we cannot undo or modify this transaction. Generally speaking, this is desirable from the point of view of the ISPs, but there are some situations when it is necessary to reclaim a block of addresses, e.g. stolen or lost keys, human error, misuse, etc. In addition, IP addresses are a finite good and must be preserved: the loss of an IP prefix impacts the whole community, as opposed to a crypto-coin: only its owner is affected.

Taking in to account that in a blockchain we can define an arbitrary set of rules, we can design some schemes to recover a block of addresses, but at the same time preserve the decentralization in any blockchain. For instance, we can let a widely recognized third party (e.g. IANA) resolve disputes between conflicting parties by issuing a special transaction that reallocates the resource. Other mechanisms are possible: time-limited allocations, multi-signature transactions, etc.

Nevertheless, the revocation approach should be agreed among the relevant players (IANA, RIRs, ISPs, institutions, etc). It is worth noting that behind these mechanisms there is a fundamental trade-off between complete centralization (traditional PKI, trust the upstream provider) and total decentralization (blockchain).

VI. IMPLEMENTATION

We have built an open-source prototype and made it publicly available[4]. We did not to fork an existing blockchain

[4] https://github.com/OpenOverlayRouter/blockchain-mapping-system

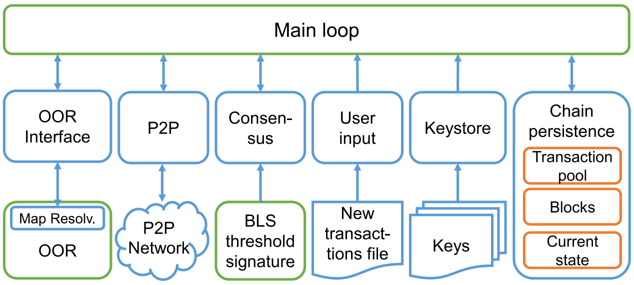


FIGURE 7. IPchain prototype architecture.

implementation since they do not fit our needs, particularly regarding the PoS consensus algorithm.

The IPchain prototype is written in Python (fig. 7), supports common blockchain operations, and implements the operations defined in section V-D for both IPv4 and IPv6 addresses. With the aim to ease user interaction, the prototype reads new transactions from a file, signs and sends them to the network, and includes a keystore to encrypt the user's private keys.

In addition, the prototype interfaces via a local socket with OpenOverlayRouter (OOR [18]). OOR is an open-source implementation of the Locator/ID Separation Protocol (LISP, RFC 6830) that creates programmable overlay network tunnels. OOR leverages IPchain to retrieve metadata related to the IP prefixes that is used in some LISP signaling messages. In what follows, we describe relevant modules of the prototype.

Data Structures: IPchain builds on Ethereum's account system, which maps pairs of blockchain addresses with the associated IP addresses. Transactions are encoded as modifications to accounts. We chose this model instead of Bitcoin's UTXO because it requires less storage and data access is easier. We modified the PyEthereum[5] Trie, DB, Utils and Transactions classes to fit our needs, and capped the block size at 2 MB.

Peer-to-Peer Network: The P2P module implements all communication functions in a broadcast-all fashion, leveraging Pyhton's Twisted[6] library for network communication. Since it does not connect all the nodes between themselves, a Distributed Hash Table[7] keeps track of the last block number, so that if a node misses some blocks it can request them.

A. PoS CONSENSUS ALGORITHM

We have leveraged part of the DFINITY blockchain PoS algorithm for our chain [19]. This chain operates a complex three-step algorithm that combines a Decentralized Random Number Generator (DRNG), a block notarization process and a finalization process. The two latter are used to resolve chain forks and achieve higher scalability. For simplicity, our prototype only uses the DRNG and does not tolerate chain forks.

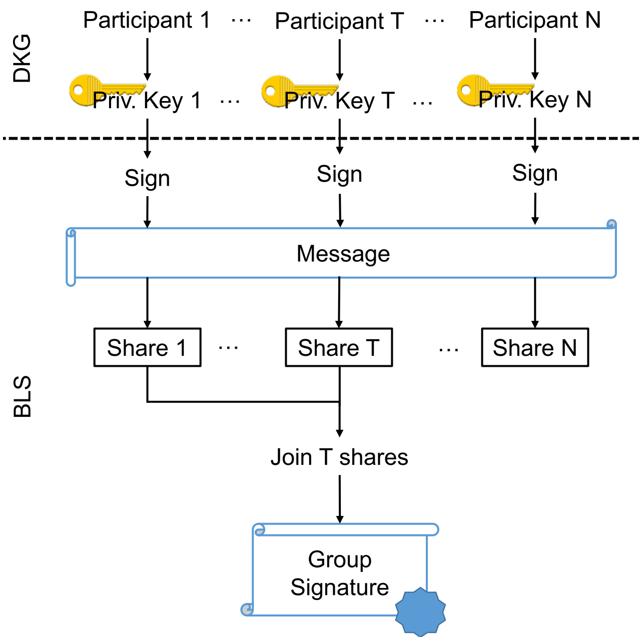


FIGURE 8. DKG (top) and BLS (bottom) operation.

The DRNG builds on the well-know topic of threshold signatures (fig. 8), in which a group of participants jointly generate a group key and a set of private keys, known as Distributed Key Generation (DKG). Later, they can use these private keys to sign a message; however, the signature is only valid if a portion of the participants (defined by a tunable parameter, the Threshold) have signed the message. The scheme we use there is the Boneh-Lynn-Shacham (BLS). The message is commonly know as *share* because it is a part of the whole signature.

In the blockchain context, we calculate the random number as the hash of the signed message (n.b. all participants sign the same message). Due to the fact that we cannot recover the signed message unless Threshold participants have signed it, we can be sure no single participant can manipulate this number. We repeat this process for each new block to get a fresh random number to select the next block signer. In addition, the Threshold signature scheme offers two key advantages: (i) since it is not necessary that *all* the participants sign the message, we can tolerate several malicious or disconnected nodes, and (ii) we can adjust the Threshold to tune the upper limit of malicious or disconnected participants depending on the current situation, as we mentioned in sec. IV-C.

Finally, our prototype also: (i) repeats the private key generation process after a predetermined number of blocks to refresh the public and private keys, and to take into account new participants that have been added in previous blocks, and (ii) in order to scale more easily, not all blockchain addresses take part in the DRNG, we select some of them with the previously generated random number.

VII. EXPERIMENTAL EVALUATION

We carried out several experiments in order to determine the applicability of IPchain in real life. We measured several

[5] https://github.com/ethereum/pyethereum
[6] https://twistedmatrix.com/trac/
[7] https://github.com/cliftonm/kademlia-1

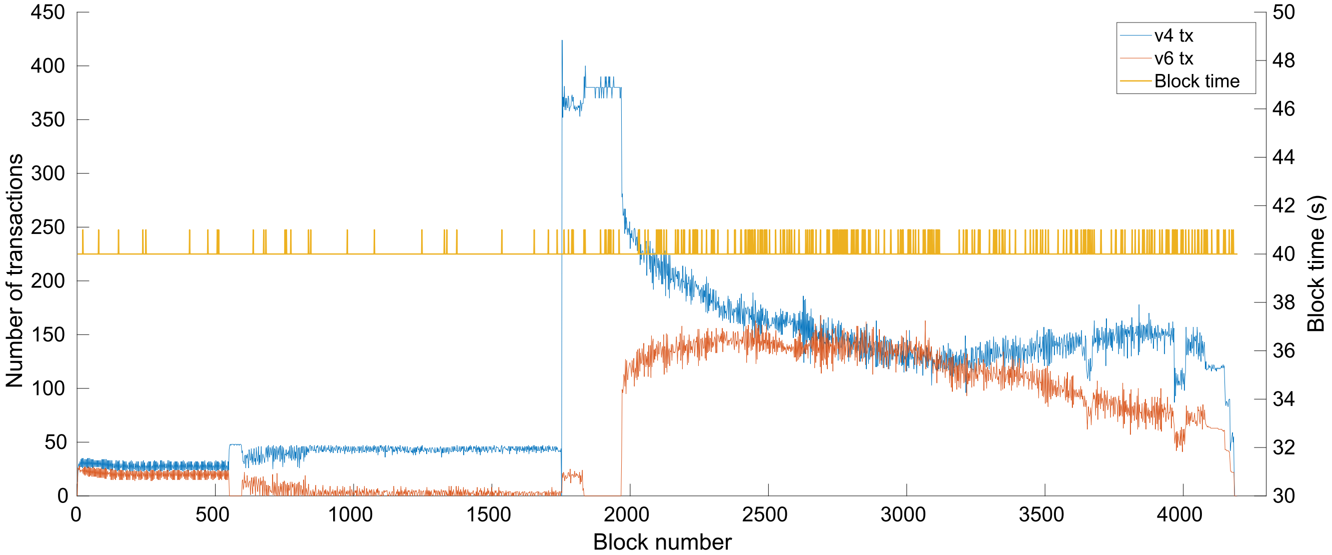


FIGURE 9. Number of transactions in each block and block time. We can see the non-uniform distribution of v4 and v6 transactions especially after block 2000.

blockchain metrics to characterize the performance and scalability of IPchain, focusing on six key metrics: (i) throughput, (ii) block time, (iii) bootstrap time, (iv) chain size, (v) block time depending on BLS threshold, and (vi) DKG refresh time. Finally, we present an analytical estimation of the required storage in the long-term.

A. BLOCKCHAIN METRICS

Throughput: For this test, we encoded in the genesis block the entire v4 and v6 address space (splitting them in large blocks of addresses, similarly to IANA's registries of v4 and v6 address space[8]). In order to simulate the typical allocation scheme of IP prefixes, we generated three groups of transactions. The first one referenced prefixes in the genesis block and split them uniformly into smaller prefixes, and the second also created smaller prefixes but referencing the transactions from the first group. Finally, we extracted the prefixes for the third level directly from the publicly available lists of the Registries' address allocations[9], with the final goal of storing real prefixes in the chain. In total, we generated around 380k transactions (82k in levels 1 and 2, 298k in level 3 - two thirds of the 430k in the RIR files).

We set up a single node (VM with two associated virtual CPUs, Intel Xeon Platinum @ 3 GHz, 4 GB RAM) with these transactions, all the associated keys and a configured DKG with 100 keys and 66% BLS Threshold, 40s block time and 2 MB block size. We can see the result of this experiment in fig. 9, which plots the number of transactions per block, separated into v4 and v6 transactions. We can see two distinct phases in the experiment: before and after block ∼1580. In this moment we increased the transaction injection speed from 1tx/s to 10tx/s. Indeed, we can see that the number of transactions per block increases from 25 to 150, approximately.

During the whole test, and especially after block 2000, we observe a high variability in the number of transactions per block, roughly spanning from 100 to 250 transactions. This is due to the non-uniform distribution of v4 and v6 transactions in the input file: the proportion of v4 and v6 transactions is not constant for the entire file. Since the node processed transactions at a fixed speed, regardless if they were v4 or v6, in a given period of time a we may not inject a constant number of v4 or v6 transactions. We can also notice this in the reduction of v4 transactions when the number of v6 increases, and viceversa.

The maximum number of transactions per block revolves around 400, so we can estimate IPchain's throughput to be approximately 10 transactions per second, in the same order of magnitude of Bitcoin. If we consider that average of number of BGP updates is between 10-15 per second [12], and that our system targets a subset of those, we can conclude that IPchain presents sufficient throughput for this application.

Blocktime: In order to verify the correct operation of the prototype, fig. 9 presents the time between each consecutive block in the right y axis. We can see that in nearly all cases it remains in the configured interval of 40 seconds, since we automatically trigger the block creation just after 40s from the timestamp of the last block. In some cases, due to data processing delays it reaches 41s.

Bootstrap test: With the aim of quantifying IPchain's cost in terms of time and compute resources, we performed a bootstrap test, i.e. adding a new node to the network and measure: (i) time to validate all the chain, and (ii) total chain storage. We must note that the time to download the blocks is negligible compared to the validation time. We used a VM with two associated virtual CPUs (Intel Xeon Platinum @ 3 GHz) and 4 GB RAM. It took 3.5 hours to verify

[8] https://www.iana.org/numbers

[9] https://www.nro.net/statistics/

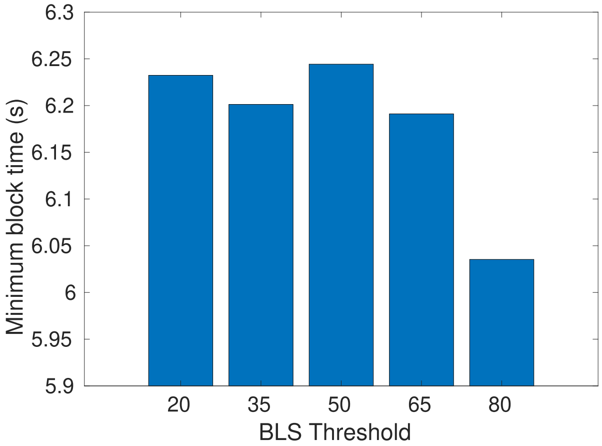


FIGURE 10. Minimum block time depending on BLS Threshold setup.

the chain, that contained 350k prefixes and required 2.5 GB of storage. This last metric lets us conclude that the chain can scale well in terms of storage for this scenario.

B. CRYPTOGRAPHY METRICS

Block time depending on BLS threshold: In this test we aim to calculate the minimum block time depending on the BLS threshold setup, which in turn determines the chain throughput. We define the minimum block time as the time it takes to generate the BLS shares, send them to the network, recover the group signature, calculate the random number and create an empty block. Fig. 10 presents the average of this time for 100 blocks for different threshold values. We set up 10 nodes in a cloud provider, each in a different region of the world, and VMs with two virtual CPUs, Intel Xeon @ 2.5 GHz, 4GB RAM. The nodes exchanged BLS shares, with 10 blockchain addresses for each (100 participants in the BLS in total). As we can see, the minimum block time revolves around 6.15 seconds regardless of the Threshold setup. Two main reasons explain this phenomenon: (i) the operation that joins the BLS shares is not computationally intensive, and (ii) the share size is too small (each share weights approximately 60 bytes) to cause an increase in the communication delay, i.e. the delay does not increase significantly if we send 80 shares instead of 20 across the network. These results suggest that the BLS Threshold does note have a noticeable impact on performance.

DKG keys renewal time: Finally, we measured how long it took to perform a key refresh depending on the number of nodes. We used the same cloud nodes than in the previous experiment. Fig. 11 presents a linear interpolation of data from three experiments. We can see that the delay grows linearly with the number of nodes, basically because in this process: (i) each node has to send a message to the rest, and (ii) the crytographic function to generate the group key is more costly than the one that joins the BLS shares.

C. LONG-TERM STORAGE ESTIMATION

We performed a storage estimation in order to evaluate IPchain's long-term feasibility. We estimated separately the number of IP prefix allocations and AS-to-prefix bindings, and made the following assumptions. For the prefix allocation, transactions of 500 bytes, an initial load of the current 600k prefixes in the BGP RIB table, plus re-allocating all BGP prefixes each year, and a growth in new prefixes exponentially adjusted to the BGP RIB table growth [20]. Regarding the bindings of AS numbers to prefixes, we assumed one AS-to-prefix binding for each block in all the /24 IPv4 address space, each transaction weights around 400 bytes, and an update rate similar to the BGP churn [21], increasing linearly each year.

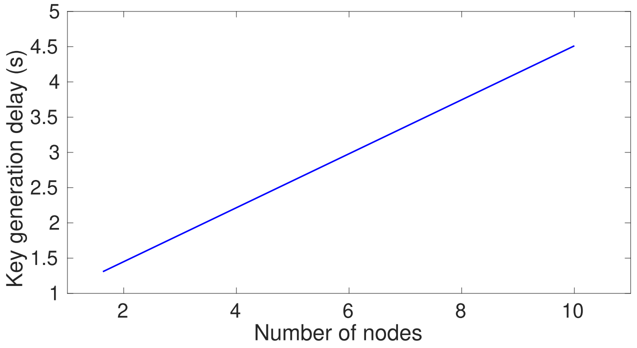


FIGURE 11. Delay to create private keys.

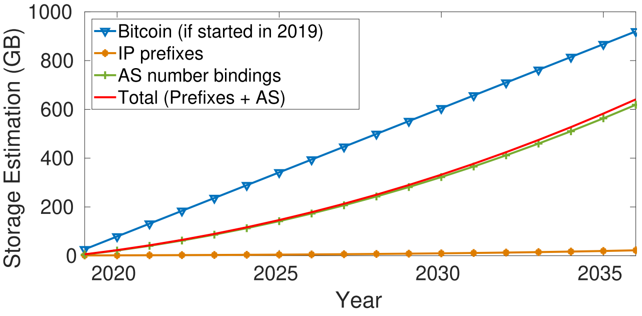


FIGURE 12. 20-year storage estimation for IPchain.

Figure 12 presents the estimation for both AS numbers and IP prefixes, their sum, and a comparison with Bitcoin, assuming it starts in 2019, keeps the 0.5 MB block size, the rate of 1 block each 10 min, and all blocks are full. We can see that prefixes account for a reduced part of the transactions when compared to AS bindings; this is due to the continuous increase in the churn and the number of bindings. Nevertheless, in 20 years' time the chain accounts for approximately 600 GB, a figure that is easily attainable by current storage systems. Finally, we can see that in this 20-year interval, we stay below the requirements of Bitcoin, which supports our thesis that a blockchain for IP addresses has moderate storage requirements in the long term.

VIII. RELATED WORK

A. BLOCKCHAIN APPLICATIONS FOR NETWORKING

Several blockchain applications oriented for networks have been proposed [22], such as the Internet of Things [23], BGP messages [24], Information Centric Networking [25] or distributed access control [26], [27]. However, the largest body of work focuses on providing naming applications with similar functions to the DNS: Namecoin[10], Blockstack [28], Ethereum Name System[11], etc. To the best of our knowledge, IPchain is the first blockchain specifically tailored for the allocation and delegation of IP addresses.

The most closely related work is [29], which advocates for an automatic IP prefix distribution system on top of an Hyperledger private blockchain, while in our proposal we maintain the manual allocation of IANA and RIRs and we use PoS instead of Hyperledger's certificate-based system. Other similar works focus on embedding also BGP path announcements in the blockchain [30], or deploying an IP allocation system as a smart contract in Ethereum [31], but neither of them focus on the benefits of PoS algorithms.

B. PoS CONSENSUS ALGORITHMS

We can find several PoS algorithms already in production systems, such as NEM's Proof of Importance [32] or Ethereum Capser. Proof of Importance is a PoS variant that takes into consideration, apart from the stake, the dynamics of the transactions (topology, frequency). Due to this, we don't find this algorithm suitable for IPchain, because our situation is not as rapidly evolving as in the NEM chain.

Regarding Ethereum Casper, we found that its punishment mechanism is not suitable for our use case (sec. V-C). Other proposed algorithms, like Algorand [33] or Ouroboros [17], are a good fit, but we were unable to find an open-source implementation.

C. SOCIAL NETWORK CONSENSUS ALGORITHMS

These kind of algorithms, such as Stellar [34], are based on a user's trust relationships with other users. In other words, users define a list of trusted nodes, and only consider transactions validated by them. Regarding our use-case, we consider that they are a promising alternative. However, since they commonly require an identification based on digital certificates, we would lose the aforementioned advantages related to simplified management.

IX. CONCLUSION

In this paper we have introduced IPchain, a Proof of Stake blockchain to store the allocation and delegation of IP addresses. We have discussed its benefits over existing systems, both in the policy side and the technical side. On the policy side, we argue that the decentralization of blockcahins can mitigate the centralization concerns of current systems, offering an alternative to distribute trust among all its participants. With respect to the technical issues, we have considered how a blockchain can ease management or offer auditabilty when compared to current solutions. Moreover, we have emphasized the advantages of a Proof of Stake consensus algorithm for our specific use case. In the end, we have evaluated the performance of our prototype storing Internet Registry data, and demonstrated that it can achieve the requirements for real-world deployments regarding throughput and long-term storage.

Possible future work revolves around adding all or part of the BGP AS-path in IPchain, so that BGP routers have additional data to verify BGP messages. Specifically, it is worth investigating how to ensure that only authorised neighbouring ASes can append a new ASN, and a scalability analysis for this new data, because the churn, throughput, and storage requirements are significantly higher. Indeed, this means storing all the active Internet BGP routes in the blockchain. On the performance side, it is worth investigating data structures for efficient management of IP prefixes in the chain (split / merge operations), and a detailed analysis on the PoS protocol scalability, e.g. how does the DKG group size affect security and block time. Finally, since IPchain does not tolerate forks, we need a reliable mechanism to select a new signer if the original one is offline or malicious.

ACKNOWLEDGMENT

The authors would like to thank J. Herrera-Joancomarti, A. Rodriguez-Donaire, J. Baylina and L. Vegoda for their helpful discussions about blockchain technology. Also M. Ferriol, E. Garcia, H. Latif, and C. Piris for their invaluable contributions to the prototype.

REFERENCES

[1] S. Goldberg, "Why is it taking so long to secure Internet routing?" *Commun. ACM*, vol. 57, no. 10, pp. 56–63, 2014. [Online]. Available: http://doi.acm.org/10.1145/2659899

[2] R. Singel. (Feb. 2008). *Pakistan's Accidental YouTube Re-Routing Exposes Trust Flaw in Net*. [Online]. Available: https://www.wired.com/2008/02/pakistans-accid/

[3] RIPE NCC. (Mar. 2019). *RPKI Certification Statistics*. [Online]. Available: http://certification-stats.ripe.net/

[4] D. Cooper, E. Heilman, K. Brogle, L. Reyzin, and S. Goldberg, "On the risk of misbehaving RPKI authorities," in *Proc. 12th ACM Workshop Hot Topics Netw. (HotNets)*, 2013, pp. 16:1–16:7.

[5] K. O'hara and W. Hall. (2018). Four Internets: The geopolitics of digital governance. Centre for International Governance Innovation, Waterloo, ON, Canada. CIGI Papers 206. [Online]. Available: https://www.cigionline.org/publications/four-internets-geopolitics-digital-governance

[6] S. Malcomson, *Splinternet: How Geopolitics and Commerce are Fragmenting the World Wide Web*. New York, NY, USA: OR Books, 2016.

[7] Y. Gilad, O. Sagga, and S. Goldberg, "Maxlength considered harmful to the RPKI," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2017, pp. 101–107.

[8] E. Heilman, D. Cooper, L. Reyzin, and S. Goldberg, "From the consent of the routed: Improving the transparency of the RPKI," in *Proc. ACM Conf. SIGCOMM*, 2014, pp. 51–62.

[9] W. George, "Adventures in RPKI (non) deployment," North Amer. Netw. Operators Group (NANOG), Baltimore, MD, USA, Tech. Rep., 2014. [Online]. Available: https://archive.nanog.org/meetings/abstract?id=2441

[10] B. Kuerbis and M. Mueller, "Internet routing registries, data governance, and security," *J. Cyber Policy*, vol. 2, no. 1, pp. 64–81, 2017.

[11] J. Paillisse, M. Ferriol, E. Garcia, H. Latif, C. Piris, A. Lopez, B. Kuerbis, A. Rodriguez-Natal, V. Ermagan, F. Maino, and A. Cabellos, "IPchain: Securing IP prefix allocation and delegation with blockchain," in *Proc. IEEE Int. Conf. Blockchain*, Jul. 2018, pp. 1236–1243.

[12] G. Huston. (Oct. 2019). *BGP Table, ASN and CIDR Reports*. [Online]. Available: http://bgp.potaroo.net/

[13] Ethereum Foundation. (Oct. 2019). *Proof of Stake FAQ*. [Online]. Available: https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ

[14] B. Awerbuch and C. Scheideler, "Robust random number generation for peer-to-peer systems," in *Principles of Distributed Systems*, M. M. A. A. Shvartsman, Ed. Berlin, Germany: Springer, 2006, pp. 275–289.

[10] https://namecoin.org/

[11] https://ens.domains/

[15] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[16] P. Daian, R. Pass, and E. Shi, "Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake," in *Financial Cryptography and Data Security*, I. Goldberg and T. Moore, Eds. Cham, Switzerland: Springer, 2019, pp. 23–41.

[17] A. Kiayias, A. Russell, B. David, and R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol," in *Advances in Cryptology*, J. Katz and H. Shacham, Eds. Cham, Switzerland: Springer, 2017, pp. 357–388.

[18] A. Rodriguez-Natal, J. Paillisse, F. Coras, A. Lopez-Bresco, L. Jakab, M. Portoles-Comeras, P. Natarajan, V. Ermagan, D. Meyer, D. Farinacci, F. Maino, and A. Cabellos-Aparicio, "Programmable overlays via open-overlayrouter," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 32–38, Jun. 2017.

[19] T. Hanke, M. Movahedi, and D. Williams, "Dfinity technology overview series, consensus system," 2018, *arXiv:1805.04548*. [Online]. Available: https://arxiv.org/abs/1805.04548

[20] G. Huston. (Jan. 2017). *BGP in 2016*. [Online]. Available: http://www.potaroo.net/ispcol/2017-01/bgp2016.html

[21] A. Elmokashfi, A. Kvalbein, and C. Dovrolis, "BGP churn evolution: A perspective from the core," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 571–584, Apr. 2012.

[22] N. Bozic, G. Pujolle, and S. Secci, "A tutorial on blockchain and applications to secure network control-planes," in *Proc. 3rd Smart Cloud Netw. Syst. (SCNS)*, Dec. 2016, pp. 1–8.

[23] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[24] A. Hari and T. V. Lakshman, "The Internet blockchain: A distributed, tamper-resistant transaction framework for the Internet," in *Proc. 15th ACM Workshop Hot Topics Netw.*, 2016, pp. 204–210.

[25] N. Fotiou and G. C. Polyzos, "Decentralized name-based security for content distribution using blockchains," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2016, pp. 415–420.

[26] D. Di Francesco Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Distributed Applications and Interoperable Systems*, L. Y. Chen and H. P. Reiser, Eds. Cham, Switzerland: Springer, 2017, pp. 206–220.

[27] J. Paillisse, J. Subira, A. Lopez, A. Rodriguez-Natal, V. Ermagan, F. Maino, and A. Cabellos, "Distributed access control with blockchain," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[28] M. Ali, J. C. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *Proc. USENIX Annu. Tech. Conf.*, 2016, pp. 181–194.

[29] S. Angieri, A. García-Martínez, B. Liu, Z. Yan, C. Wang, and M. Bagnulo, "A distributed autonomous organization for Internet address management," *IEEE Trans. Eng. Manage.*, to be published.

[30] I. Sfirakis and V. Kotronis, "Validating IP prefixes and AS-paths with blockchains," 2019, *arXiv:1906.03172*. [Online]. Available: https://arxiv.org/abs/1906.03172

[31] Q. Xing, B. Wang, and X. Wang, "BGPcoin: Blockchain-based Internet number resource authority and BGP security solution," *Symmetry*, vol. 10, no. 9, p. 408, 2018.

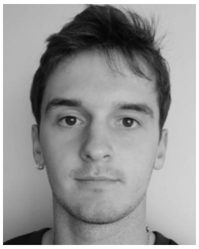
[32] NEM Foundation. (Feb. 2018). *NEM Technical Reference*. [Online]. Available: https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf

[33] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proc. 26th Symp. Oper. Syst. Princ.*, 2017, pp. 51–68.

[34] D. Mazieres, "The stellar consensus protocol: A federated model for Internet-level consensus," Stellar Develop. Found., San Francisco, CA, USA, Tech. Rep., 2015. [Online]. Available: https://dl.acm.org/citation.cfm?doid=3341301.3359636

JORDI PAILLISSE received the degree in telecommunications engineering from UPC, in 2013, where he is currently pursuing the Ph.D. degree in computer architecture. He has been a Visiting Student with École Polytechnique Fédérale de Lausanne, Switzerland. His main research interests include the future Internet architectures, overlay networks, blockchain applications for networking, and software-defined networking.

JAN MANRIQUE received the B.Sc. degree in computer science from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 2020. He is especially interested in the next-generation blockchain consensus algorithms, especially Proof of Stake and Federated Byzantine Agreement. He currently develops firmware for the IoT devices.



GUILLEM BONET received the B.Sc. degree in computer science from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 2019. He has been a Visiting Student with TU Delft. He is currently working in the private sector developing Cloud Solutions. His research interests revolve around blockchain performance and incentive structures for Proof of Stake algorithms.

ALBERTO RODRIGUEZ-NATAL received the B.Sc. degree in computer science from the University of Leon, Spain, in 2010, and the M.Sc. and Ph.D. degrees from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 2012 and 2016, respectively. He has also been a Visiting Researcher with the National Institute of Informatics, Japan, in 2014. He joined Cisco in 2016, where he continues his research on new network architectures with special focus on software-defined networking and programmable overlay networks.

FABIO MAINO received the M.S. degree *(Laurea)* in electronic engineering and the Ph.D. degree in computer and network security from Politecnico di Torino, Italy. He is currently a Distinguished Engineer with Cisco Systems, in the Chief of Technology and Architecture Office, where he leads a team that focuses on driving innovation on network virtualization and SDN. He has about 50 patents issued or filed with the U.S. PTO, and has contributed to various standardization bodies, including the IEEE, IETF, and INCITS.

ALBERT CABELLOS is currently an Associate Professor with the Department of Computer Architecture, UPC. He has been a Visiting Professor with Cisco Systems, San Jose, CA, USA, KTH, Kista, Sweden, Agilent Technologies, Edinburgh, U.K., the Massachusetts Institute of Technology, Cambridge, MA, USA, and the University of California at Berkeley, Berkeley, CA, USA. He has participated in several research projects funded by companies (Cisco, Intel, and Samsung) as well as publicly funded (FP7, H2020, NSF, and a national funding agency). He is also a Co-Founder of the Open Overlay Router Open Source Project.