Internet Routing Blockchain: an Hyperledger Fabric consortium blockchain for Internet Routing Registries

Stefano Angieri
University Carlos III of Madrid
Spain
sangieri@pa.uc3m.es

Marcelo Bagnulo
University Carlos III of Madrid
Spain
marcelo@it.uc3m.es

Alberto García-Martínez
University Carlos III of Madrid
Spain
alberto@it.uc3m.es

***Abstract*—The Internet Routing Registry (IRR) is a distributed routing database that provides a mechanism for validating the contents of BGP announcement messages and mapping an origin AS number to a list of networks [1]. The data stored in the IRRs contains information on the business relationship between the ASes which can be used to perform route leaks prevention, as described in ASIRIA [2]. Due to the distributed nature of the IRRs, the quality of the information available can suffer from inconsistency across different IRRs and information stall-ness. Besides, the complexity of the RPSL syntax limits the precision of ASIRIA relationship inference algorithm. In this paper we present the Internet Routing Blockchain (IRB) an implementation of the IRR functionality with Hyperledger Fabric. The IRB relies on a permissioned blockchain technology that is inherently distributed, allows to preserve the decentralised nature of the IRR, overcomes the centralized governance model limitation of ASPA employing a consortium based model, provides consistency and information stall-ness prevention and offers a simple declaration syntax for AS relationships.**

***Index Terms*—Blockchain, Hyperledger Fabric, Internet Routing Registries , Interdomain Routing, BGP, Security, Route Leaks, Routing Policies.**

I. Introduction

A. Introduction

The Internet Routing Registry (IRR) is a globally distributed routing information database, established in 1995 with the purpose of ensuring the stability and consistency of Internet-wide routing by sharing information between network operators. IRRs are distributed repositories, individually operated by different organizations, where ASes declare their routing policies. This information can be used to perform route leak detection and prevention. According to the current Mutually Agreed Norms for Routing Security (MANRS) guidelines [3], operators are encouraged to use the information available in the IRR to create filters that prevent route leaks. In [2], we have motivated and described how ASIRIA can use the information stored in IRRs to prevent and detect route leaks serving as a bootstrap for ASPA while ASes start to register their relationship policy or their client list in RPKI servers. In short each AS registers its routing policies in the IRR. When an AS receives a route, the router of the AS uses this information, according to the ASIRIA specification, to validate it. ASIRIA takes advantage of the valley-free properties of route propagation in the internet, so that it can determine if the route is a leak or not. This, combined with origin validation, provided by the RPKI, results in quite good protection. Using the IRRs may allow to bootstrap this process and provide the ability to validate a significant number of routes even when the ASes are not yet actively registering their relationships in RPKI servers. However, there are some limitations to overcome. In this section, we will present the shortcoming of these two solution, that will serve to motivate the work on the Internet Routing Blockchain (IRB) that we present in this paper.

1) ASIRIA limitations: When designing and implementing ASIRIA, we identified two main limitations about using the data available in the IRR to prevent route leaks, namely, the quality of the data and the limitations of the relationship inference algorithm. We describe them next. Regarding the quality of the information available in the IRRs, we identified two issues. First, the information may be *stalled* and second the information may be *inconsistent across the different IRRs*. Regarding stall-ness, the ASes sometimes fail to keep updated the routing policy information they registered. Regarding consistency, there are 34 IRRs and it we have observed that the information in them may diverge for the same routing object. Indeed, as each IRR runs independently, ASes may register their routing policies in different IRRs. Sometimes, this is the result of an ASes changing the reference IRR it normally uses to update their policies. When an AS starts using a new IRR, it may not remove the information in the IRR it was using previously and with time, this information may be stalled and inconsistent with the information available in the newly used IRR. We analyzed the data available in all the IRRs in March 2020 and we found that there are 67k aut-num objects, of which 5% of them are present in more than one IRR. Given that it is unlikely that one AS maintains its routing policy updated in several IRRs, these duplicated records are likely to be or become stalled. By relying on the blockchain technology, we are able to guarantee consistency across the different IRB operators by design. Regarding the limitation of the inference algorithm which goal is to determine the ASes relationship based on their routing policy, as we presented when designing ASIRIA, we are able construct inference algorithms that have a high precision. Unfortunately, we are unable to achieve 100% precision. The fundamental reason for this is that we need to infer the relationship from the routing policy declared in the aut-num objects in the IRR. While it is possible to design the IRB to support the RPSL language [4], in order to maximize the inference algorithm precision, we design the IRB so that the types of relationships are explicitly declared by the ASes in the registry. Specifically, we design the IRB to store records where an AS can declare that a relationship with another AS exists, and explicitly declare the type of this relationship. Actually p2p and p2c/c2p are by far the most common types of relationships [5]. There are other types of relationships between ASes (such as siblings, partial transit and hybrid relationships) but they are much less common, so we focus the design of the IRB in p2p and p2c/c2p types of relationships. To this end, the IRB allows ASes to explicitly declare p2p, p2c and c2p relationships. Supporting sibling relationships is trivial, because, as opposed to partial transit and hybrid relationships, siblings are defined for the whole AS. On the other hand, partial transit results in different relationships on a per prefix granularity and hybrid relationships implies that two ASes have different relationships on different interconnection points.

2) ASPA limitations: ASPA explicitly allows ASes to declare its providers in the RPKI. As such, it addresses both limitations identified above, namely consistency of the data and difficulties in the inference of the relationships. The consistency is guaranteed because there is a single RPKI database. Relationships are explicitly declared in the ASPA record, so no inference is required. ASPA only allows the declaration of c2p relationships, but this is enough for the vast majority of the cases. The ASPA limitations are related to the governance model. As ASPA is part of the RPKI, this means that all ASPA records are under the control of the RPKI hierarchy and subject to errors, abuses and misuses, suffering from the *jurisdictional overflow* described in [6]. Furthermore several concerns have been raised about the proposal to move RPKI registry into a cloud architecture [7], [8] in order to improve performance, availability, and reduce costs. These concerns are about the loss of autonomy, independence and responsibility and the legal or political risks, and contrasts with the current IRR decentralised structure, which we argue is worthwhile preserving. Moreover in [9] has been remarked that the application of blockchain technology to the IRR system with unforgeability, distributed consensus, and provable timeline characteristics open the possibility to address many data governance problems in routing security. The IRB blockchain can be set up in a customizable decentralized and permissioned fashion to address the above concerns. The IRB consortium can be started by a subset of the involved players such as RIRs, NIRs, ISPs or any interested organization. All IRB nodes maintain a copy of the shared ledger as shown in the high-level architectural view in figure 1. As a new transaction to register or modify a policy is performed on the system, the ledgers of every peer get synchronized. Besides instead of having a decentralized set of autonomous databases, we ensure to have an unique decentralized shared ledger to maintain updated information.

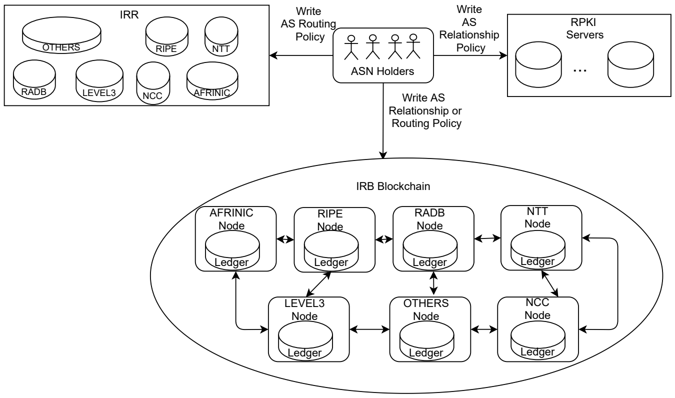

Fig. 1: High-Level Architectural View

Relying on blockchain technology that is inherently distributed, the IRB preserve by design the decentralised nature of the IRR, provides consistency and information stall-ness prevention and allow AS owner to store their relationship policy in a shared registry. Besides the permissioned model assure that only certified resource holders are able to store their information in the IRB.

II. Design of the IRB

In this section we describe the design of the proposed IRB solution.

1) Blockchain technology: As motivated earlier, we build the IRB using blockchain technology. We decided to use a permissioned blockchain framework, more specifically Hyperledger Fabric (HF) [10]. HF is an open source enterprise-grade permissioned distributed ledger technology platform which provides an open and modular architecture and plug-and-play components to accommodate a wide range of use cases. Considered as a de facto standard for enterprise blockchain platforms [11], HF is used for private and consortium blockchain. In a permissioned blockchain, the consortium, i.e., the set of organization maintaining the network, have the control over the consortium admission management while every organizations have the control over its client's membership management. Using HF we can build a consortium blockchain where every entity or organizations interested in its maintenance is allowed to join. For our experiments we decide to simulate a set up of a consortium blockchain maintained by RIRs. The reason behind that choice are revealed in section III.

The usage of a consortium blockchain overcomes several limitation that comes using a public blockchain network [12] regarding the exposure of sensitive information related to resources and the resource control. It may seem contradictory to the use of a public blockchain with controlling the exposure of the information, but a consortium permissioned blockchain allows defining different levels of exposure for different information sets, e.g., personal information. The IRB data model can be designed to be both private or/and public accessible. As example it is possible to design the data model exposure regarding sensitive data to be private while the relationship information can be public exposed and can be easily accessed and used for validation. Besides private data collections [13] provide the opportunity to customize the restricted subset of organizations allowed to access that data. About the usage of a public blockchain to manage resources that are critical for the functioning of the Internet, a potential threat is the loss of the keys that bind these resources. The use of a permissioned blockchain where the organizations are in charge of the certificate issuance, renewal and revocation [14], as every certificate is bind to a specific resource, provides the same level of control over the managed resource as the one provided by current solutions. Besides the usage of a permissioned consortium model brings a better control over the disputes that may raise in case of an eventual attempt to tamper the system. Indeed the knowledge of the participants identities together with the high level of traceability offered by blockchains simplifies the eventual dispute resolution. Moreover only well known actors can take part to the blockchain maintenance where each actor has the same right to vote despite the number of resources it manages. Besides the consortium can decide to disconnect a malicious player.

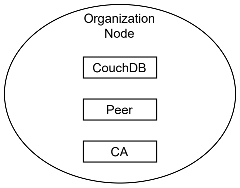

Fig. 2: Organization Node

The IRB blockchain is composed of a number of network components, run by independent *organizations*, and of an ordering service node. As shown in figure 2, every organization willing to participate in the IRB has to run the following HF components: a Certificate Authority, at least one peer and a couchDB server. The Certificate Authority is in charge of identity registration, enrolment certificate (X.509 certificates are used for the authorization and authentication) issuance, renewal and revocation [14]. Every organization has its own Certificate Authority by which it manages the identities of its clients. The Peer is the main network component which participates in the consensus, hosts ledgers and smart contracts [15]. The Couchdb is a nosql alternate state database that allows to model data on the ledger as JSON and offers ”rich queries” against data values rather than being restricted to queries to the keys [16]. CouchDB allows data retrieval via simple HTTP requests. All the organizations willing to join the IRB are at least required to deploy the above mention network components. These organisations are the ones running the blokchain. In addition, there are clients of these organisations that are the ones issuing transactions that in turn populate the blockchain. For the specific case of the IRB, the model based on peers and their clients allows mapping naturally the roles of the current IRR, meaning that entities currently operating an IRR will become peers in the IRB while entities registering routing policies in the IRR will become clients in the IRB. This means that we expect that the peer organisations will include RIRs, NIRs, some large ISPs and other entities which purpose is solely to run an IRB instance (similar to the ones existing today running an IRR).

While the service providing ordering to blockchain transactions is in general a critical component, in this particular case it is not. The reason is that the altering the order of registration of different relationships does not have any significant impact. So, we propose that the ordering system is operated by the different peer organisations in a round robin fashion a customizable time slot each. Whenever needed, there are already existing solution to set up a Byzantine Fault Tolerance (BFT) ordering service [17], [18] within HF.

2) Membership management: Being a permissioned blockchain, the IRB must manage its membership for both the organisations composing the IRB and their clients. We describe them next.

a) Organisation membership management: Organisations are the ones running the network components of the blockchain, storing its data, validating and endorsing the transactions. The consortium, composed of the organizations bootstrapping the IRB, is in charge of managing organizations identities and membership. This means, that it is the consortium of organisations who decide to accept new peers. As example a new organization must be approved by at lest n/2 +1 of the total set of organizations to join the IRB. The specifics about how this is done is out of the scope of this paper. The bootstrapping of the consortium requires an initial set of organisations to agree to build the IRB, for which a good candidate set would be a subset of the RIRs as it has been simulated in this paper but this is also out of the scope of this paper.

b) Client membership management: Any entity holding an AS number can become a client of the IRB. Only legitimate holders of the AS number resources can register relationships involving this AS number (ASN). This naturally implies that the IRB membership is tightly related to ownership of AS numbers. We consider two different situations of an entity holding an AS number. The first situation is when the ASN holder has an RPKI certificate associated to the ASN [19]. In this case, the ASN holder can prove that it is the legitimate holder of the ASN by exhibiting control of the private key associated to the public key contained in the certificate. So, any entity having an RPKI certificate should automatically be accepted as a client in the IRB and be allowed to issue transactions affecting that ASN. In particular, each entity having an RPKI certificate attesting the ownership of an ASN issues become a client of the peer organization ran by the RIR who issued the said certificate. The second situation is when the holder of the ASN has obtained it from a RIR/NIR but does not have an RPKI certificate. In this case, the entity is client of the RIR that allocated the ASN and it is the RIR that validates that the allocation is legitimate using its own means. In this case, the client will obtain a certificate from the peer organisation certification authority to be able to issue transactions in the IRB.

3) Asset description: IRB Blockchain network participants perform *ASN-pair contract transactions* to achieve their business objectives such as the registration of the business relationship between two ASes. The ASN-pair contract defines the ASN_PAIR asset, an object characterized by a list of states, namely *Issued*, *Verified*, *Invalid*, and of a set of transactions that trigger events and state transitions such as *Issue*, *Sign* and *Invalid*. A security level is assigned to each state of the asset. The security level is the number of recognized interaction between certified ASN holders on an specific mutual relationship. In detail securityLevel=1 is associated to *Issued* state, securityLevel=2 to *Verified* state and securityLevel=0 to *Invalid* state. The ASN_PAIR asset life cycle is defined by the finite state machine in figure 3.

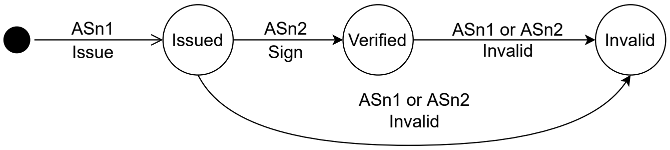

Fig. 3: Life Cycle of Holdership Attestation

The ASN_PAIR record is finally described by its set of fields:

- State
- Autonomous System Number 1
- Autonomous System Number 2
- Relationship
- securityLevel

Each ASN_PAIR record is accessible by the primary keys [ASN1,ASN2] and [ASN2,ASN1].

We now describe a sample scenario, depicted in figure 4, where the holder of ASN1 declares the business relationship established with the holder of the ASN2 using the IRB blockchain.

The ASN1 holder performs an initial *Issue* transaction. The contract checks the existence of an ASN_PAIR record with the same keys [ASN1,ASN2] and [ASN2,ASN1]. If keys do not match with an existing record a new ASN_PAIR record is created. Its securityLevel is set to 1 and its state to *Issued*; If keys match an existing record, the contract allow its modification only if its state is set to *Invalid*. Otherwise the contract raises an error. Once the ASN_PAIR with keys [ASN1,ASN2] has been issued, the ASN1 holder can only invalid the record performing an *Invalid* transaction which sets the record security level to 0 and state to *Invalid*. Meanwhile the ASN2 holder has the right to perform or an *Invalid* transaction or a *Sign* transaction. The *Sign* transaction sets the ASN_PAIR record security level to 2 and state to *Verified*, in the means that has been publicly recognized that both parties involved in the business relationship have officially signed the described relationship. At this stage, whether ASN1 or ASN2 holder wants to modify the relationship, he has to invalid first the previous record and then re-issue the record with the modified information. Hence this record has to be signed again by the counterparty to increase its security level to 2.

Even if an information with an higher security level is always preferred, to perform leak prevention it is possible to use an ASN_PAIR record which security level is 1. In the specific case where an AS agrees with the declarations of its neighbors, it may not need to sign its record. However, this mechanism provides means to easily invalidate the declarations of other party.

4) Third Party Data Retrieval: A third party who wants to retrieve the information stored in IRB can accomplish the task in two different manners: contract call or via simple http requests. The maximum number of records that can be requested in one contract call is limited to 100 thousand for security reason. In case there is the need to retrieve n*100 thousand records at same time, it is possible to use pagination [20] to split the n request or to perform a simple http request. We designed the IRB to query data via http requests which in turn is faster and has no limitation regarding the number of records to be requested in a single operation. Besides an interested third party can sing an event subscription to the IRB to receive all the event generated by the contract triggered by transactions. Using an event based model to manage contract transactions provides an easy traceability of the incremental differences generated by the newly transaction insertion.

III. Dataset and experiments

We provide an interactive prototype of the IRB blockchain [21]. Tests to show that the contract is functionally correct are also available at [21]. In this section we show results regarding the prototype scalability. We study in detail the effect in data retrieval time and the amount of memory required as the number of relationships registered grow.

1) IRB Scalability: We first describe the dataset that has been used for testing the IRB blockchain's scalability. The dataset contains information regarding the AS relationships inferred by CAIDA [22], which represents a subset of the Internet's relationship, combined with the IANA's data regarding the ASN assignments to RIRs [23], in figure 5. The resulting dataset is the information regarding the AS relationship depicted per RIR. As this information is divided in 5 organizations and it comes from data that are mainly considered as a Ground truth, we can simulate a close-to-real scenario for testing the IRB blockchain as a consortium run by RIRs. The total number of unique ASN counted is about 72K and it represents the total number of users as clients that can participate in the IRB blockchain at the time of this writing. The number of new relationships in figure 6, depicted per RIR for year, is the amount of information that has to be inserted to bootstrap the system with the current available information. The number of changed relationship, depicted per RIR for year in figure 7, is the amount of information that has suffered changes from year to year and changes are

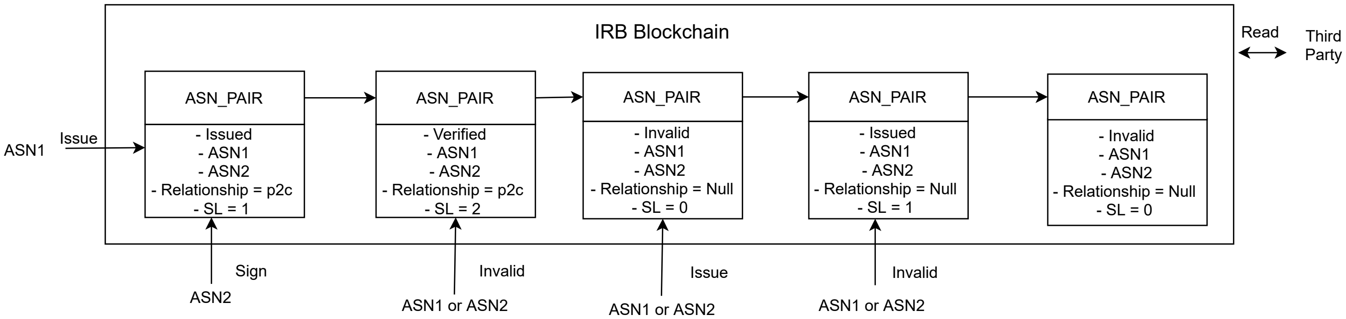

Fig. 4: Interaction with Blockchain

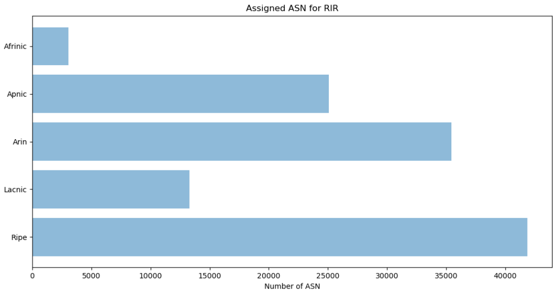

Fig. 5: ASN assignments

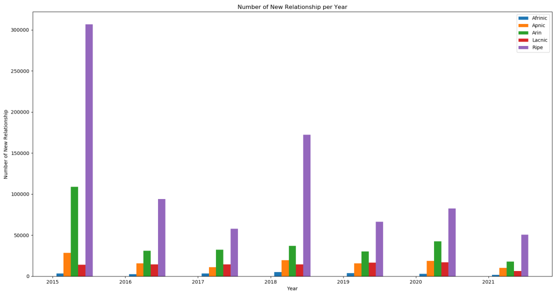

Fig. 6: New relationships per year

an estimation of the modifications that has to be registered. The total amount of records to be inserted to simulate the actual state of ASes information, based on Caida and IANA data, is about 915 K of records. As each new record requires 200 Bytes, this result in at least 200 Megabytes of data. As expected, the time of information retrieval and the couchDB memory size are directly proportional to the number of records and they increase linearly. Respectively the time to retrieve data is comprised between 0.1 and 14 seconds while the the CouchDB memory size between 0 and 200 Megabytes.

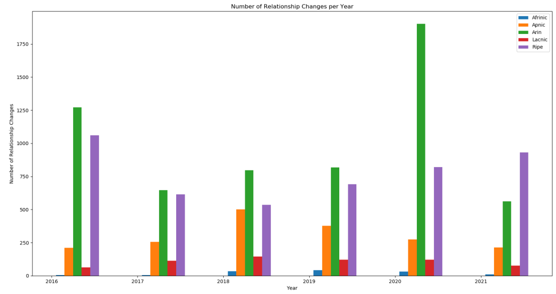

Fig. 7: Relationship changes per year

2) *Record insertion:* In this section we report an experiment regarding the insertion of record as batches. To measure the scalability we needed to populate the IRB blockchain within the whole information available. As a matter of time we decide to insert the information as grouped records batches. When inserting new records into the IRB Blockchain, the bottleneck is the round trip time between the application and the CouchDB. The CouchDB delays a fixed amount of time to create the asset which varies from 0.1 seconds and increases linearly with the size in bytes of the record to be inserted [24]. An ASN_PAIR record has a size of 200 bytes. Due to the previously described CouchDB limitation, is crucial to to limit and fix the number of transactions that are spammed to the network in a defined time window, in way to avoid the overload of the involved network components. We decide to spam a rate of 10 transactions per second. In the first experiment we tried to insert a full batch of 40 thousand relationships at a fixed rate of 10 tx per second within one process run in a client application. In figure 8 is it possible to see how the confirmation time performance degrades while the number of spammed records increases. The confirmation time peak is reached at the number of 28 thousand records. At this point we can see, apparently, that the confirmation time seems to be constant, but we have verified that all those transactions are actually discarded as the receiver queue of incoming transaction is full. To overcome the previously described limitation, we have decided to use a batch of 2500 transactions for process with a fixed rate of 10 tx per second and to run a new process for every bulk of records. Using this configuration it is possible to see, figure 9, that we can maintain the confirmation time of 10 tx constant between 1 and 2 seconds, constrained by the theoretical limit given by the asset creation time of the couchDB. Based on the constructed dataset, we can expect a close-to-real scenario where the 915

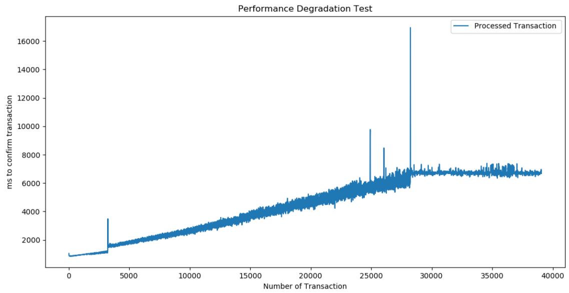

Fig. 8: Insertion performance degradation single process

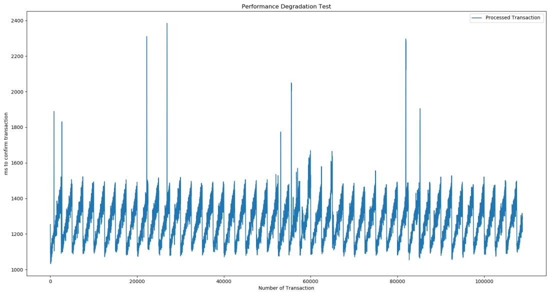

Fig. 9: Insertion performance degradation multi process

K records are divided in 6 years resulting in 417 records per day or 17 records per hour. As shown in the previous experiment, under certain conditions, the IRB can maintain a constant record insertion time of 10 tx between 1 and 2 seconds which is independent of the total number of records previously inserted in the system. Besides the IRB can easily support the hypothetical charge of 17 transaction per hour and its eventual increase.

IV. Related Work

In this section we consider previous research work on the application of blockchain technology to route leak prevention. In [9] authors compare IRRs to other methods of governing routing data in a way that enhances internet security, such as RPKI and BGPSec and Routing Security. Besides they consider and remark the opportunity offered by the application of blockchain technology in the data governance problem associated with routing, actually pushing the research in that direction. ISRchain [25] is a blockchain-based inter-domain secure routing framework to manage Internet resources, implemented on Quorum test network using the Raft consensus algorithm. In ISRChain each BGP speaker is required to run a blockchain node. RPL chain [26] is a privacy-preserving route leak protection and trusted execution blockchain based environment implemented on Microsoft Azure. In RLPchain, each AS maintains a global confidential and tamper-proofing inter-domain routing policy repository, using cryptography primitives and detecting and preventing inter-domain routing leaks. Miquel et al. [27] proposes a decentralized approach to prevent routing leaks based on the routing policy repository stored in an HF blockchain. However Miquel's work provides an architectural model in which each AS is required to run an HF peer. All works [25]–[27] provide a prototype which design architecture requires the run of one blockchain node for each AS. Instead we propose a lighter architectural model where ASN holders can work as clients of the subset of organization that are in charge of running peers. Indeed we argue that offering the opportunity to join the blockchain as a client to those ASes holder with limited interest in running run a node, would facilitate the adoption of the service resulting in the improvement of routing security. Besides such architectural model based on peers and their clients allows mapping naturally the roles of the current IRR to the IRB.

V. Conclusion

In this paper, we have presented the Internet Routing Blockchain, an HF consortium blockchain based solution to preserve the distributed nature of IRRs which provides information consistency and stall-ness prevention by design. The data stored in IRB can be used by novel route leaks prevention solution such as ASPA and ASIRIA. The experimental evaluation shows that the prototype scales linearly with respect to the amount of information to be processed. Also, we have simulated a close-to-real scenario for the prototype bootstrap.

References

[1] Merit. Overview of the irr. http://www.irr.net/docs/overview.html. [Online; accessed 25-Jun-2021].

[2] eUnivrsity Carlos III de Madrid. Practicable route leak detection and prevention with asiria. *Submitted*.

[3] MANRS. MANRS Implementation Guide, Filtering. https://www.manrs.org/isps/guide/filtering/. Accessed: 2020-11-11.

[4] David Kessens, Tony J. Bates, Cengiz Alaettinoglu, David Meyer, Curtis Villamizar, Marten Terpstra, Daniel Karrenberg, and Elise P. Gerich. Routing Policy Specification Language (RPSL). RFC 2622, June 1999.

[5] Peyman Faratin, David D Clark, Steven Bauer, William Lehr, Patrick W Gilmore, and Arthur Berger. The growing complexity of internet interconnection. *Communications & strategies*, (72):51, 2008.

[6] Stefano Angieri, Alberto Garcia-Martinez, Bingyang Liu, Zhiwei Yan, Chuang Wang, and Marcelo Bagnulo. A distributed autonomous organization for internet address management. *IEEE Transactions on Engineering Management*, 67(4):1459–1475, 2020.

[7] Felipe Victolla Silveira. Rpki repositories and the ripe database in the cloud. https://labs.ripe.net/author/felipe_victolla_silveira/rpki-repositories-and-the-ripe-database-in-the-cloud/. [Online; accessed 9-Jul-2021].

[8] Felipe Victolla Silveira, Kaveh Ranjbar, Daniel Karrenberg, Fergal Cunningham, Vesna Manojlovic, and Antony Gollan. Ripe ncc and the cloud: Let's start again. https://labs.ripe.net/author/felipe_victolla_silveira/ripe-ncc-and-the-cloud-lets-start-again/. [Online; accessed 9-Jul-2021].

[9] Brenden Kuerbis and Milton Mueller. Internet routing registries, data governance, and security. *Journal of Cyber Policy*, 2(1):64–81, 2017.

[10] Hyperledger Community. Hyperledger fabric. https://www.hyperledger.org/use/fabric. [Online; accessed 30-Jun-2021].

[11] IBM. What is hyperledger fabric? https://www.ibm.com/topics/hyperledger. [Online; accessed 6-Jul-2021].

[12] Marco Hogewoning. A review of blockchain applicability to internet number resources. https://labs.ripe.net/author/marco_hogewoning/a-review-of-blockchain-applicability-to-internet-number-resources/. [Online; accessed 6-Jul-2021].

[13] Hyperledger Community. Private data. https://hyperledger-fabric.readthedocs.io/en/release-2.2/private-data/private-data.html. [Online; accessed 9-Jul-2021].

[14] Hyperledger Community. hyperledger-fabric-ca documentation. https://readthedocs.org/projects/hyperledger-fabric-ca/downloads/pdf/latest/. [Online; accessed 31-May-2021].

[15] Hyperledger Community. Peers. https://hyperledger-fabric.readthedocs.io/en/release-2.2/peers/peers.html. [Online; accessed 31-May-2021].

[16] Hyperledger Community. Couchdb as the state database. https://hyperledger-fabric.readthedocs.io/en/release-2.2/couchdb_as_state_database.html. [Online; accessed 31-May-2021].

[17] Jeonghyeon Ma, Yongrae Jo, and Chanik Park. Peerbft: Making hyperledger fabric's ordering service withstand byzantine faults. *IEEE Access*, 8:217255–217267, 2020.

[18] João Sousa, Alysson Bessani, and Marko Vukolic. A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 51–58, 2018.

[19] Dr. Charles W. Lynn Jr., Karen Seo, and Stephen Kent. X.509 Extensions for IP Addresses and AS Identifiers. RFC 3779, June 2004.

[20] Hyperledger Community. Query the couchdb state database with pagination. https://hyperledger-fabric.readthedocs.io/en/release-2.2/couchdb_tutorial.html#cdb-pagination. [Online; accessed 9-Jul-2021].

[21] Stefano Angieri. Irb_blockchain. https://github.com/steang91/IRB_Blockchain. [Online; accessed 9-Jul-2021].

[22] V. Giotsas, M. Luckie, B. Huffaker, and k. claffy. Inferring Complex AS Relationships. In *ACM Internet Measurement Conference (IMC)*, pages 23–30, Nov 2014.

[23] IANA. Autonomous system (as) numbers. https://www.iana.org/assignments/as-numbers/as-numbers.xhtml. [Online; accessed 6-Jul-2021].

[24] N.K. Lincoln. Hyperledger fabric 1.4.0 performance information report. https://hyperledger.github.io/caliper-benchmarks/fabric/resources/pdf/Fabric_1.4.0_javascript_node.pdf. [Online; accessed 31-May-2021].

[25] Di Chen, Yang Ba, Han Qiu, Junhu Zhu, and Qingxian Wang. Isrchain: Achieving efficient interdomain secure routing with blockchain. *Computers & Electrical Engineering*, 83:106584, 2020.

[26] Jiarui Yue, Yajuan Qin, Shuai Gao, Wei Su, Guobiao He, and Ningchun Liu. A privacy-preserving route leak protection mechanism based on blockchain. In *2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE)*, pages 264–269, 2021.

[27] Miquel Ferriol Galmés, Roger Coll Aumatell, Albert Cabellos-Aparicio, Shoushou Ren, Xinpeng Wei, and Bingyang Liu. Preventing route leaks using a decentralized approach. In *2020 IFIP Networking Conference (Networking)*, pages 509–513, 2020.