Securing Route Origin Authorization with Blockchain for Inter-Domain Routing

Guobiao He, Wei Su, Shuai Gao, Jiarui Yue
School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China
17111014@bjtu.edu.cn, wsu@bjtu.edu.cn, shgao@bjtu.edu.cn, 18120168@bjtu.edu.cn

***Abstract*—The inter-domain routing with BGP is highly vulnerable to malicious attacks, due to the lack of a secure means of verifying authenticity and legitimacy of inter-domain routes. Resource Public Key Infrastructure (RPKI) is a new security infrastructure to verify that an IP address block holder has authorized an Autonomous System (AS) to originate routes by maintaining a Route Origin Authorization (ROA) repository, preventing the most devastating prefix hijacks in BGP. However, RPKI is a centralized hierarchical architecture that may empower the centralized authorities to unilaterally revoke or compromise any IP prefixes under their control. To eliminate the risks of RPKI, we present ROAchain, a novel BGP security infrastructure based on blockchain. Different from RPKI, ROAchain is a decentralized architecture, in which each AS maintains a globally consistent and tamper-proof ROA repository, authenticating the legitimacy of route origin and preventing BGP prefix hijacks. In ROAchain, a novel consensus algorithm is proposed to guarantee the strong consistency, scalability, and security of the system. Moreover, an incremental deployment scheme is designed without changing the current BGP protocol. Finally, ROAchain is implemented in Golang and validated on the Google Cloud.**

***Index Terms*—BGP security; ROA; decentralized; tamper-proof; blockchain**

I. Introduction

The inter-domain routing with BGP is a ”default-accept” architecture [1]: any autonomous system (AS) can originate a BGP routing announcement for any IP prefix and other ASes will accept the BGP announcement by default. This makes BGP vulnerable to routing attacks and the most common is prefix hijacks [2]. The RPKI [3] is a BGP security infrastructure to prevent prefix hijacks by maintaining a ROA repository. The RPKI establishes a top-down hierarchy of authorities that are rooted at the Regional Internet Registries (RIRs), which provides a trusted mapping from an IP prefix to a set of autonomous systems (ASes) that are authorized to originate this prefix.

Although the RPKI is very effective against prefix hijacking, it still faces several fundamental risks and challenges. First, RPKIs hierarchical architecture empowers centralized authorities to unilaterally revoke authorization or take down IP prefixes under their control [4], [5]. Ethan Heilman improves the transparency of RPKI through entitling parties to consent to revocations of their IP address space [1], however, the centralized hierarchical architecture is still reserved and the risks remain. Moreover, centralized authorities are also an easy target to impose censorship or information control by state actor [6]. Second, the global RPKI only provides a loosely consistent view. There are five RIRs, issuing the certificate and ROAs independently [7]. Thus, one AS router may have a different view about a particular prefix than another since there is no unified RPKI repository. Third, the management of certificates in RPKI is complex [8]. Once the root certificate is updated, all the certificates issued by it must also be updated.

Annex to ISBN 978-3-903176-28-7 2020 IFIP

Recently, several blockchain-based schemes are proposed as an alternative BGP security infrastructure to RPKI [9]–[11]. These blockchain-based solutions eliminate the potential risks of unilateral revocation, misconfigured, or legally compelled to misclassify a legitimate BGP route as bogus in RPKI. However, they also introduce some new challenges to the current BGP. First, the throughput of current blockchain-based solutions is very low (e.g. The transaction throughput of SBTM, BGPcoin, and IPchain are 7, 15, and 10 transactions/sec respectively.), and can't deal with the peak load of ROA registration, update, and revocation when IPv6 is deployed on a large-scale in the future [12]. Second, blockchain-based solution itself will introduce new security risks [13]. For example, SBTM is based on Proof-of-Work (PoW), while BGPcoin and IPchain are based on Proof of Stake (PoS). Both of PoW and PoS are vulnerable to Sybil attack and easy to fork, resulting in data inconsistency [14], [15]. Finally, all these solutions lack the consideration of a compatible deployment scheme.

To cope with the risks of RPKI and challenges of existing blockchain-based solutions in securing ROA, this paper presents ROAchain. ROAchain is a decentralized architecture, providing means of associating a cryptographic key with an IP address block and greatly simplifying the certificate management. In ROAchain, each AS maintains a consistent and tamper-proof ROA repository through consensus. To reduce the impact of blockchain-based solution on the current BGP performance and enhance the security of blockchain itself, a novel consensus algorithm is presented. Moreover, an incremental deployment scheme is designed without changing the current BGP protocol, greatly reducing the complexity and difficulty in deployment. The experiment results show that ROAchain outperforms the current blockchain-based solutions BGPcoin, SBTM, and IPchain in terms of consensus latency, throughput, security, scalability, and compatibility.

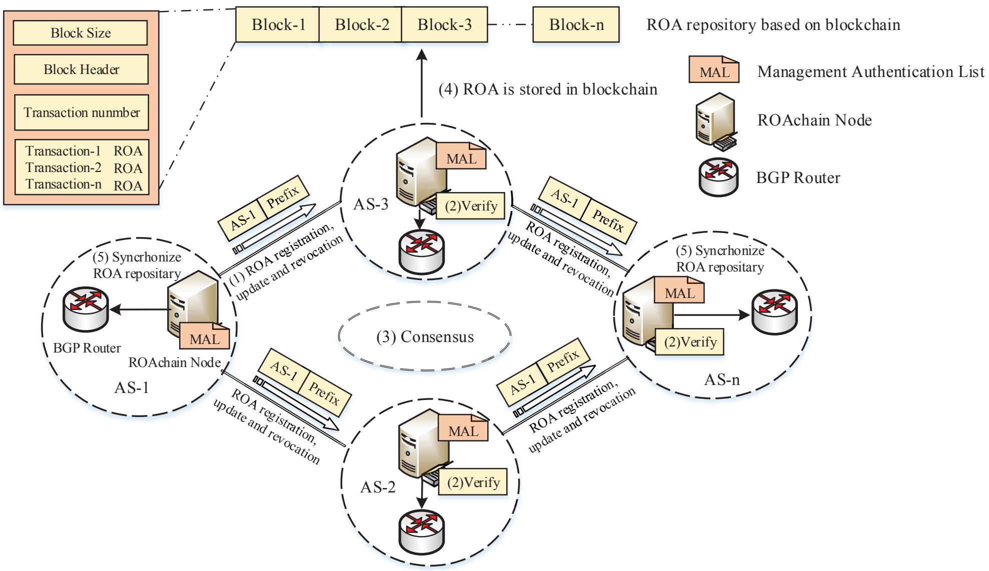

Fig. 1. ROAchain architecture

II. ROAchain architecture

ROAchain consists of four components as follows. A management authentication list (MAL) is an access control mechanism to prevent Sybil attacks, which contains a white list of ROAchain nodes' public key and its corresponding AS information. A ROAchain node added in each AS is used to initiate ROA operations and authenticate the legitimacy of the BGP route. It also translates the blockchain-based ROA into normal BGP update message. A tamper-proofing and consistent ROA repository based on blockchain records all the previous ROA operations, including registration, update, and revocation. A novel consensus algorithm is presented to minimize the impact of blockchain to the current BGP and enhance the security of the system.

The workflow of ROAchain is illustrated in Fig. 1.

(1) In ROAchain, a ROAchain node is added in each AS. It initiates ROA operations that belonged to its own AS such as registration, update, and revocation, which contains the binding relationship of IP prefixes and its corresponding AS.

(2) Each AS verifies the legitimacy and correctness of the ROA operations based on the assignment information of RIRs or its own monitoring information.

(3) After the verification, all ASes will make a consensus about ROA operations, which will be stored in the blockchain.

(4) Each ROAchain node maintains a global consistent ROA repository based on blockchain, containing all the previous ROA operations records.

(5) Finally, each BGP router synchronizes the ROA repository from a ROAchain node, verifying the legitimacy of BGP route origin and preventing prefix hijacking. A validated prefix will be stored in the BGP routing table.

A. ROA operations of ROAchain

ROA operation is used to register, update or revoke a ROA, which is stored in the blockchain as a transaction. Unlike RPKI using maxLength, which is vulnerable to forged-origin subprefix hijacks [16], ROAchain uses minimal ROAs.

1) ROA operation format: ROA=SIGi [Trans-type, version, asID, ipAddrBlocks, Trans-fee, Timestamp, Exp-time].

There are three types of ROA operations identified by *Trans-type*, including registration, update and revocation. The same with [17], *Version* number of the ROA MUST be 0; *asID*field contains the AS number that is authorized to originate routes to the given IP address prefixes; *ipAddrBlocks* field encodes the set of IP address prefixes to which the AS is authorized to originate routes. *transaction fee* is added as the cost of ROA operation. The accurate generation time of transactions is recorded by *Timestamp* to prevent replay attacks. *Exp-time* is the a valid period of ROA information.

2) Block format of ROAchain: B_r *= [Type, r, V,* C_i, Q_{r-1}, $H(B_{r-1})$, M_e, S_h, M_h, T_s, $SIG_i(H(B_r))$, T_x*].*

There are two types of blocks identified by *Type*, including the final block and sharding block. Here, *r* is the round number and *V* is used to distinguish the version of the block specification. The C_i is the credence value to quantify the credibility of ROAchain node. A nonce Q_{r-1} is used to guarantee the security of consensus process. Each block contains the hash $H(B_{r-1})$ of the previous block, in which Merkle root hash M_e of transactions is adopted to guarantee the data integrity. The hash S_h and M_h are used to ensure integrity of the latest state of the ROA repository and management certification list respectively. The accurate generation time of a block is recorded by a timestamp T_s, preventing replay attacks. The block signature $SIG_i(H(B_r))$ is used to verify the identity of block producer. All the ROA operations are recorded in blockchain as transactions T_x.

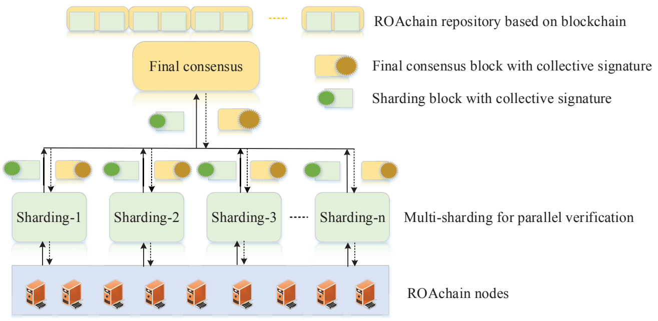
ROAchain repository based on blockchain
Final consensus
Final consensus block with collective signature
Sharding block with collective signature
Sharding-1
Sharding-2
Sharding-3
Sharding-n
Multi-sharding for parallel verification
ROAchain nodes

Fig. 2. Design of consensus algorithm in ROAchain

B. Design of consensus algorithm

The throughput of current blockchain-based solutions for BGP can't deal with the peak load of ROA operations when IPv6 is deployed on a large-scale in the future. Besides, blockchain-based solutions will introduce new security risks such as forking, Sybil attacks, etc. Therefore, a novel consensus algorithm is designed in ROAchain to not only improve the ROA operation performance but also ensure the security of blockchain-based solution.

The design of our consensus algorithm is illustrated in Fig. 2, which has the following features. First, multi-sharding and collective signature are introduced to improve the performance of consensus [18], [19]. As blockchain is based on TCP communication, which is subject to MITM-attacks, DoS-attacks, and network partitioning attacks, penetrating the resulting communication and message dissemination delays. The sharding members and final consensus members are randomly selected from all ROAchain nodes to participate in the consensus process, mitigating DoS-attacks, and network partitioning attacks. Moreover, transport layer security (TLS) can be used to prevent MITM-attacks. Second, credence value is presented to quantify the credibility of the ROAchain node. The leader is randomly selected based on its credence value, reducing the probability of malicious ROAchain node being selected as a leader.

1) Sharding consensus: The transactions are processed in parallel in each sharding, producing the sharding blocks with collective signatures.

$$\frac{j-1}{s} \leq .H\left(SIG_i\left(r, \mathrm{S}, C_i, Q_{r-1}, H\left(B_{r-1}\right)\right)\right) \leq \frac{j}{s}, 1 \leq j \leq s \quad (1)$$

$$\frac{j-1}{s} \leq .H\left(SIG_i\left(r, \mathrm{S}, C_i, Q_{r-1}, H\left(B_{r-1}\right)\right)\right) \leq \frac{j}{s} * p, 1 \leq j \leq s \quad (2)$$

$$p = \frac{N_s * s}{n} \quad (3)$$

The ROAchain nodes are evenly and randomly assigned to different shardings based on (1). Then, N_s sharding members are randomly selected from all ROAchain nodes based on (2) to participate in the sharding consensus. Here, $N_s \geq 2f{+}1$, f is the max faulty number of ASes that can be tolerated in each sharding. The consensus executed in rounds r. S indicates that this is the sharding member selection process to distinguish from the final consensus process. C_i is the credence value of the ROAchain node i. Q_{r-1} is the nonce in the last round, guaranteeing the randomness of consensus member selection process and preventing target DoS attacks. In (3), p represents the probability being selected as a sharding consensus member. j is the sequence number of the sharding. The s and n is the total number of shardings and ROAchain nodes respectively.

The sharding leader is in charge of assembling the transactions and producing sharding blocks with a collective signature. Its selection process is based on the equation (4).

$$\frac{j-1}{s} \leq .H\left(SIG_i\left(r, \mathrm{S}, C_i, Q_{r-1}, H\left(B_{r-1}\right)\right)\right) \leq p_j^l + \frac{j-1}{s} \quad (4)$$

$$p_j^l = \frac{\tau}{N_s} \quad (5)$$

p_j^l in (5) refers to the probability of a sharding member being selected as the leader. Based on the law of large numbers in probability statistics, $p_j^l = 1/N_s$ can ensure that there is exactly one leader being selected when N_s is large enough. However, there may be none is selected when N_s is small. To prevent this from happening, the parameter τ is introduced to guarantee that at least one but no more than τ potential sharding leader is selected. If several potential leaders are selected simultaneously in a sharding, the one with the smallest hash will be the leader. Then, each leader assembles all transaction belonged to its sharding into one sharding block. Here, ROAchain uses BonehLynnShacham (BLS) [20] to produce a collective signature.

2) Final Consensus: The final consensus is used to guarantee the consistency of the ROA repository, aggregating all sharding blocks into one final block.

$$.H\left(SIG_i\left(r, C, C_i, Q_{r-1}, H\left(B_{r-1}\right)\right)\right) \leq p_c \quad (6)$$

$$p_c = \frac{N_c}{n} \quad (7)$$

The final consensus members are selected based on (6). C indicates that this is a final consensus election process and p_c is the probability of ASes being selected as a final consensus member. In (7), the number of final consensus member is N_c and $N_c \geq 3f{+}1$. The final consensus leader is selected based on (8). The probability p^l of a ROAchain node being selected as a final consensus leader is based on its credence value C_i and the total credence value S_c in the system.

$$.H\left(SIG_i\left(r, C, C_i, Q_{r-1}, H\left(B_{r-1}\right)\right)\right) \leq p^l \quad (8)$$

$$p^l = \frac{C_i}{S_c} \quad (9)$$

$$C_i = C_i + 1 \quad (10)$$

$$S_c = \sum_{i=1}^{n} C_i \quad (11)$$

$$Q_r = \mathrm{H}(SIG_i(Q_{r-1}, r)) \quad (12)$$

The leader is the one with the smallest hash if there is more than one potential leader being selected. Equation (10) illus-

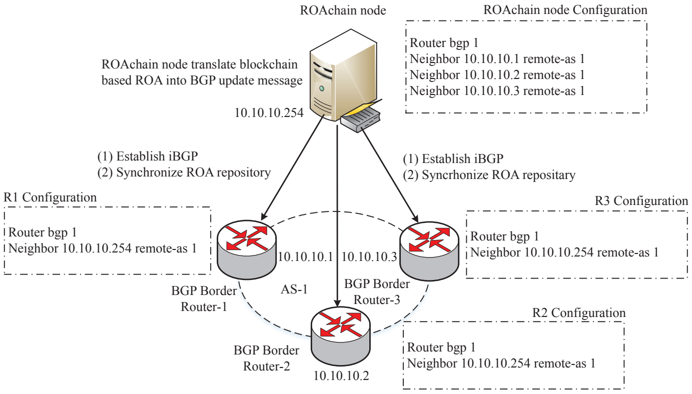
ROAchain node
ROAchain node Configuration
Router bgp 1
Neighbor 10.10.10.1 remote-as 1
Neighbor 10.10.10.2 remote-as 1
Neighbor 10.10.10.3 remote-as 1
ROAchain node translate blockchain based ROA into BGP update message
10.10.10.254
(1) Establish iBGP
(2) Synchronize ROA repository
(1) Establish iBGP
(2) Syncrhonize ROA repositary
R1 Configuration
Router bgp 1
Neighbor 10.10.10.254 remote-as 1
R3 Configuration
Router bgp 1
Neighbor 10.10.10.254 remote-as 1
10.10.10.1
10.10.10.3
BGP Border Router-1
AS-1
BGP Border Router-3
R2 Configuration
Router bgp 1
Neighbor 10.10.10.254 remote-as 1
BGP Border Router-2
10.10.10.2

Fig. 3. Compatibility design of ROAchain

trates the credence value increasing process. In (12), Q_r is a nonce in the next round. The final consensus leader assembles all sharding blocks into a final block with a collective signature and broadcasts it to the network. After received and verified the final block, the system will turn into the next round.

III. Compatibility design

ROAchain has good incremental deployment and compatibility capabilities without changing the current BGP protocol, only requiring changes to BGP router configuration. There is a ROAchain node added in each AS, which is in charge of the ROA registration, update, and revocation. Each ROAchain node maintains a fully consistent view of the ROA repository based on blockchain and establishes iBGP protocol with all BGP routers within an AS. Besides, a data conversion module is added in the ROAchain node, translating the blockchain-based ROA into a normal BGP update message. The route in BGP router can be classified as valid, not found, and invalid. The compatibility design and workflow is illustrated in Fig. 3.

(1) The ROAchain node and all BGP routers within an AS will establish iBGP neighbor relationships with each other.

(2) A data conversion module translates the blockchain-based ROA information into normal BGP update messages, broadcasting the latest incremental ROA information to all iBGP neighbors.

(3) The received IP prefixes in each BGP router is marked as valid, invalid or not found according to the synchronized ROA repository and the routing policy. The valid IP prefix will be added into the BGP routing table, preventing BGP from prefix hijacking.

IV. Implementation and Evaluation

We implement the prototype of ROAchain in Golang, using the gRPC for network communication, BLS Collective Signing and the SHA-256 hash function. The prototype is deployed on the Google Cloud using 20 virtual machines (VMs). Each VM is configured with 4 vCPU, 15G memory, simulating 50 ROAchain nodes. We measure the latency and network bandwidth between VMs using a network performance measurement tool Iperf. The latency between VMs is about 2.10 ms, the bandwidth is about 1.90 Gbits/sec on average. The total number of ROAchain nodes varies from 100 to 1000. The number of each sharding and final consensus members are fixed in 100 and 150 respectively.

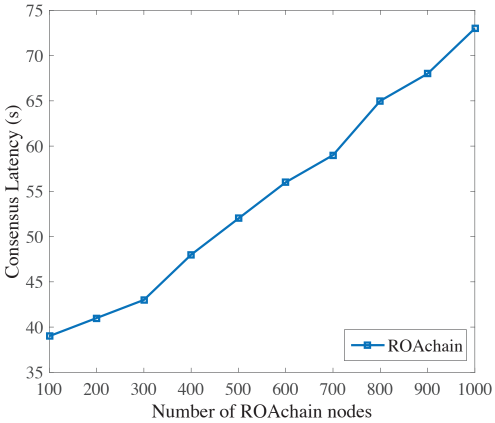

Fig. 4. Consensus latency of ROAchain

A. Consensus latency of ROAchain

Fig. 4 illustrates the consensus latency of ROAchain, which increases only a little as the sharding number increasing. This is because the transactions are processed in parallel in each sharding. Hence, ROAchain is high scalability, which is crucial to cope with the increasing of AS number with the using of 32-bits AS number in the future. Currently, the publication of new ROAs in the largest repositories of RPKI takes about 10-15 minutes [7]. The consensus latency of ROAchain is less than 80 seconds, much less than the update time interval in RPKI. Therefore, ROAchain can ensure that the ROA operation can be conducted almost in real-time.

B. Throughput of ROAchain

The throughput is shown in Fig. 5, which increases almost linearly as the sharding number increasing due to parallel transaction processing in each sharding. Currently, the average of BGP update rate is between 10-15 per second and the hourly peak BGP update rate is roughly 1,000 [12]. The throughput of ROAchain is large than 25 with 1 sharding and up to 140 transactions per second with 10 shardings under the current configuration. If we simulation 5 nodes in each Virtual Machine with 4 vCPU and 15G memory, the throughput will be 10 times of this, reaching 1400 transactions per second. Therefore, our architecture ROAchain is high scalability and can deal with the peak BGP update rate.

C. Comparison of ROAchain and other blockchain-based solutions for inter-domain routing

As is shown in Table I, ROAchain outperforms current blockchain-based solutions such as SBTM, BGPcoin, and IPchain in terms of consensus latency, throughput, security, scalability, and compatibility. For consensus latency, BGPcoin and IP chain are based on PoS, which needs to wait for extra 7 blocks of time (105s) for final confirmation [15]. Therefore,

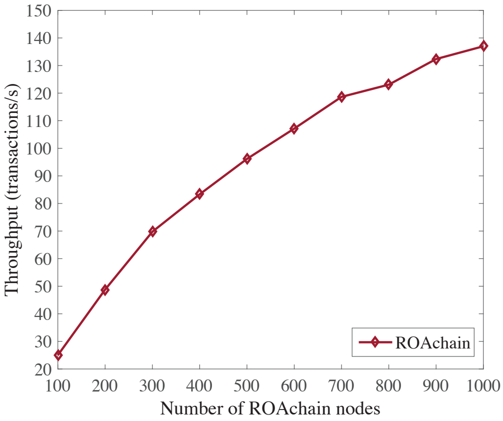

Fig. 5. Throughput of ROAchain

ROAchain outperforms BGPcoin and IPchain. The throughput of ROAchain increases almost linearly as the sharding number increasing. For security, a novel consensus algorithm is designed in ROAchain, which is non-forking and high security, while the consensus of SBTM, BGPcoin, and IPchain is easy to fork. In terms of scalability, the ROAchain has good scalability both in the number of nodes and throughput, while current SBTM, BGPcoin, and IPchain can only scale in the number of nodes. Besides, ROAchain proposes a compatible scheme, which needs little change of current BGP router.

Conclusion and Future Work

In this paper, we present a novel BGP security infrastructure based on blockchain, called ROAchain, eliminating the security risks of RPKI and the challenges of existing blockchain solutions in securing ROA. A novel consensus algorithm is designed to minimize the impact of blockchain-based solution to the current BGP and enhance the security of the system. Moreover, a compatible scheme is proposed to reduce the complexity and difficulty in deployment, which needs no changes in the current BGP protocol. Our experiment results show that ROAchain outperforms current blockchain-based solutions such as SBTM, BGPcoin, and IPchain in terms of consensus latency, throughput, security, scalability, and compatibility.

For future work, we plan to extend our work to AS path security and route leak protection (RLP), providing a comprehensive solution for BGP security based on blockchain.

Acknowledgment

This work is supported by ”Science and Technology Winter Olympics” key projects under Grant No.2019YFF0303101, National NSF of China under Grant No. 61972026, and New Generation Information Technology Innovation Project under Grant No. 2019J02020.

TABLE I
Comparison of ROAchain and other blockchain-based solutions for inter-domain routing

	this paper	[9]	[10]	[11]
Content	ROAchain	SBTM	BGPcoin	IPchain
Consensus latency(s)	39-73	600	15	40
Throughput (transactions/s)	25-136	7	7-15	10
Security	high	medium	medium	medium
Scalability	high	medium	medium	medium
Compatibility scheme	yes	no	no	no

References

[1] E. Heilman, D. Cooper, L. Reyzin, and S. Goldberg, “From the consent of the routed: Improving the transparency of the rpki,” in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 51–62.

[2] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford, “A survey of bgp security issues and solutions,” *Proceedings of the IEEE*, vol. 98, no. 1, pp. 100–122, 2009.

[3] R. Bush and R. Austein, “The resource public key infrastructure (rpki) to router protocol, version 1,” 2017.

[4] D. Cooper, E. Heilman, K. Brogle, L. Reyzin, and S. Goldberg, “On the risk of misbehaving rpki authorities,” in *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*. ACM, 2013, p. 16.

[5] M. Wählisch, R. Schmidt, T. C. Schmidt, O. Maennel, S. Uhlig, and G. Tyson, “Ripki: The tragic story of rpki deployment in the web ecosystem,” in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*. ACM, 2015, p. 11.

[6] R. Subramanian, “The growth of global internet censorship and circumvention: A survey,” *Communications of the International Information Management Association (CIIMA)*, vol. 11, no. 2, 2011.

[7] R. Bush, “Origin validation operation based on the resource public key infrastructure (rpki),” 2014.

[8] A. PHOKEER, “Interdomain routing security: motivation and challenges of rpki,” *Timss Technical Report*, 2013.

[9] A. de la Rocha Gómez-Arevalillo and P. Papadimitratos, “Blockchain-based public key infrastructure for inter-domain secure routing,” in *International Workshop on Open Problems in Network Security (iNetSec), volume IFIP eCollection-1 of Open Problems in Network Security, Rome, Italy*, 2017, pp. 20–38.

[10] 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.

[11] J. Paillisse, J. Manrique, G. Bonet, A. Rodriguez-Natal, F. Maino, and A. Cabellos, “Decentralized trust in the inter-domain routing infrastructure,” *IEEE Access*, vol. 7, pp. 166 896–166 905, 2019.

[12] G. Huston, “Bgp table, asn and cidr reports,” http://bgp.potaroo.net/, [Online].

[13] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, “A survey on security and privacy issues of bitcoin,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3416–3452, 2018.

[14] S. Nakamoto *et al.*, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[15] V. Buterin, “A next generation smart contract & decentralized application platform (2013) whitepaper,” *Ethereum Foundation*.

[16] Y. Gilad, O. Sagga, and S. Goldberg, “Maxlength considered harmful to the rpki,” in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. ACM, 2017, pp. 101–107.

[17] M. Lepinski, D. Kong, and S. Kent, “A profile for route origin authorizations (roas),” 2012.

[18] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, “Enhancing bitcoin security and performance with strong consistency via collective signing,” in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 279–296.

[19] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, “A secure sharding protocol for open blockchains,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 17–30.

[20] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weil pairing,” in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2001, pp. 514–532.