Computer Networks 217 (2022) 109362

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

RouteChain: Towards Blockchain-based secure and efficient BGP routing ☆

Muhammad Saad, Afsah Anwar, Ashar Ahmad, Hisham Alasmary, Murat Yuksel, David Mohaisen *

University of Central Florida, United States of America

ARTICLE INFO

Keywords:
Blockchain
Autonomous systems
BGP
Distributed systems

ABSTRACT

Routing on the Internet is defined among autonomous systems (ASes) based on a weak trust model where it is assumed that ASes are honest. While this trust model strengthens the connectivity among ASes, it also results in an attack surface that can be exploited to hijack the routing paths. One such attack is known as the BGP prefix hijacking, in which a malicious AS broadcasts IP prefixes that belong to a target AS, thereby hijacking its traffic. In this paper, we propose *RouteChain*: a blockchain-based secure BGP routing system that counters BGP hijacking and maintains a consistent view of the Internet routing paths. Towards that, we leverage provenance assurance and tamper-proof properties of blockchains to augment trust among ASes. We group ASes based on their geographical (network) proximity and construct a bi-hierarchical blockchain model that detects false prefixes prior to their spread over the Internet. We evaluate *RouteChain* using three different consensus protocols and show its effectiveness by drawing a case study with the Youtube hijacking of 2008. Our results show that *RouteChain* can efficiently detect false prefix announcements and prevent BGP attacks over the Internet.

1. Introduction

Data flows on the Internet between autonomous systems (ASes), which are connected in a peer-to-peer (P2P) topology [1,2]. Typically, an AS represents a collection of Internet Protocol (IP) prefixes, and to which data is routed [3]. Due to P2P architecture, ASes require an effective communication protocol to maintain updated routing paths and monitor network changes. The semantics of this communication among ASes are determined by a standardized routing protocol called the Border Gateway Protocol (BGP) [4]. Although efficient in practice, BGP is broadly based on a weak trust model, whereby ASes assume that their neighboring ASes behave honestly and propagate correct routing information. However, sometimes interests of ASes (i.e., networking operators managing them) can be in conflict, and this weak notion of trust can be breached by malicious ASes. As a result, this opens the door for attacks whereby malicious ASes may attempt to hijack the traffic of a target AS. A well-known attack that captures this exploitation of trust is known as the "BGP prefix hijacking" or simply the BGP hijacking [5,6].

In BGP hijacking, a malicious AS announces fake (impersonated or unreachable) IP prefixes that belong to a target AS. Conforming to the expectations of honesty and trustworthiness, neighboring ASes accept those prefixes and modify their routing paths for reachability. They further propagate the prefixes to their neighboring ASes. As the BGP announcement traverses across ASes, they update their routing tables according to the values in the announcement. Eventually, as the routing converges, the traffic destined for the legitimate AS – an AS that owns the announced prefix – gets diverted to the malicious AS, thereby causing a traffic hijacking. The redirected traffic does not reach the correct destination, thereby causing revenue and reputation losses.

Although various solutions have been proposed to preventing the BGP prefix hijacking attacks, including BGPSec and RPKI, those solutions have not been deployed widely in practice [7], making these attacks possible for the Internet's critical functionality of routing. A key reason behind this is that these solutions take a clean-slate approach towards redesigning routing protocols and policies, by introducing additional infrastructure for their operation. However, due to the capital invested in the existing Internet infrastructure, ASes and ISPs are reluctant to migrate towards these solutions despite the known security threats and their clear benefits.

☆ This manuscript is an extension of an earlier work that appeared in IEEE International Conference on Blockchain and Cryptocurrency (ICBC 2019) (Saad et al., 2019). This work is supported by Air Force Material Command award FA8750-16-0301 and Global Research Lab program of the National Research Foundation NRF-2016K1A1A2912757.

* Corresponding author.
E-mail addresses: saad.ucf@knights.ucf.edu (M. Saad), afsahanwar@Knights.ucf.edu (A. Anwar), ashar@cs.ucf.edu (A. Ahmad), hisham@Knights.ucf.edu (H. Alasmary), Murat.Yuksel@ucf.edu (M. Yuksel), mohaisen@cs.ucf.edu (D. Mohaisen).

https://doi.org/10.1016/j.comnet.2022.109362
Received 16 March 2022; Received in revised form 7 September 2022; Accepted 10 September 2022
Available online 16 September 2022

Embracing the policy-based functional challenges and the security risks in routing, we propose a new framework, *RouteChain*, to counter BGP attacks by using blockchains. Blockchains have introduced secure and robust ways of augmenting trust in distributed systems. Through secure-by-design protocols, blockchains enable tamper-proof data management without the need of a trusted intermediary. ASes broadly reflect a distributed mesh of interconnected systems that often lack consensus over correct protocol execution. It is therefore intuitive to bring blockchain to the design space of ASes' management to upgrade their security while maintaining operational consistency. *RouteChain* explores the usefulness of blockchains for BGP.

In *RouteChain*, we treat a BGP announcement as a transaction, and record each announcement in a blockchain ledger shared among the peers (ASes). To maintain a consistent blockchain, ASes execute a consensus protocol which specifies the *correct* blockchain state and prevents malicious announcements from propagating in the network. Blockchains use various consensus protocols to maintain consistency. In *RouteChain*, we use Proof-of-Stake (PoS), Proof-of-Authority (PoA), and Proof-of-Elapsed Time (PoET) [8–10] due to their high fault tolerance and low transaction confirmation time. We develop a blockchain testbed to evaluate the performance of *RouteChain* in the wild [11]. Our testbed consists of nodes hosted across different ASes and we evaluate the time they take to validate the correctness of a new BGP announcement.

The premise of our work relies upon a consensus agreement among ASes, prior to the launching of an attack. To that end, we contrast the timings of a known BGP attack on Youtube, with the consensus time in *RouteChain*. Our results show that while all consensus protocols meet the baseline requirement of preventing the attack, in particular, *Clique* (a PoA-based protocol) achieves the lowest transaction confirmation time and the maximum transaction throughput. *RouteChain* is incrementally deployable and backwards compatible with the current operations of ASes. ASes can use it in parallel with their current routing policies as an additional security feature. Therefore, *RouteChain* does not require ASes to switch from legacy systems to a new protocol paradigm.

Contributions. We make the following contributions: 1. We introduce a blockchain-based routing framework called *RouteChain* that prevents BGP hijacking. 2. We set up a testbed in different ASes across the world and evaluate *RouteChain*'s performance under three blockchain consensus protocols. 3. Our experiments show that *RouteChain* can neutralize a BGP hijack attempt in all three consensus protocols. Among them, *Clique* achieves the best performance with minimum transaction confirmation latency. 4. We show that unlike other proposals to counter BGP attacks, *RouteChain* can be easily implemented alongside existing protocols used by ASes.

Organization. The rest of the paper is organized as follow. In §2, we provide the background and preliminaries of our work. In §3, we introduce the problem statement, the threat model, the motivation, and the methodology. In §4, we present our proposed solution *RouteChain*, along with its design and analysis. In §5, we present the experiments and results. In Section 6, we discuss the advantages and limitations of our work. That is followed by related work and conclusion in §7 and §8.

2. Background and preliminaries

This section briefly reviews the background of ASes, BGP, and blockchains, aligned with the scope of this work.

2.1. Autonomous systems and BGP

ASes. The Internet is composed of interconnected entities that forward data from a source to a destination. These entities can range from small local area network (LAN) switches, ASes that connect geographically distributed communication devices and networks.

An AS is a collection of connected routers whose IP prefixes are assigned to an Internet Service Provider (ISP). These routers adhere to the routing policy defined by the ISP. Every AS on the Internet is assigned a unique Autonomous System Number (ASN), which is used as an identifier in inter-AS, a.k.a. inter-domain, routing. An AS is responsible for routing the traffic among networks hosted by itself and other networks hosted by its neighboring ASes. For that, an AS uses the Interior Gateway Protocol (IGP) to enable communication among its internal routers attached to its hosted networks, and the Exterior Gateway Protocol (EGP) to reach the routers in the neighboring ASes. Inter-AS relationships are established according to economic or business relationships among ISPs owning ASes. The relationship between neighboring ASes can be peering or customer–provider engagement, and typically necessitates a level of trust between them.

BGP. The Border Gateway Protocol (BGP) is a standard protocol that is designed to connect edge routers between two neighboring ASes. BGP enables the propagation of reachability and routing information from one AS to another [12]. It is also used for routing information within the AS itself. BGP operates in two ways: It uses interior BGP (iBGP) for routing among the network peers within an AS, and the external BGP (eBGP) for routing to/from other ASes [12,13]. In this paper, we are concerned with eBGP that pivots communication among edge routers of neighboring ASes. Fig. 1 shows the connection of routers within and outside the ASes. The bold circles represent the edge routers that are connected to either the internal or external routers through the BGP protocol.

BGP Attacks. The exploitation of trust among neighboring ASes leads to BGP attacks. These attacks can be broadly classified into two types: partial attack and complete attack [14]. The partial attack occurs when an adversarial AS announces an identical IP prefix as that of the victim AS. One such attack was launched on Youtube on February 24th, 2008, by an ISP that owned AS17577 [15,16]. AS17577 started to announce the prefix 208.65.153.0/24 – which actually belonged to the Youtube AS36561 – to its upstream provider AS3491. AS3491 further propagated the prefix to its neighboring ASes, leading to the Youtube hijacking. In the partial hijacking, the adversary has no significant advantage over the victim. Since the two announcements are the same, therefore when any AS receives the announcement, it can either switch to it or continue with the old routing path based on the path length.

In complete attacks, the adversarial AS announces more specific prefixes than the target AS. Since the default forwarding scheme is based on longest prefix matching, ASes switch to more specific prefixes. When an adversary announces more specific prefixes, any AS that receives the announcement inevitably switches to the new routing path. Therefore, in this attack, the adversary has a significant advantage over the victim [14]. The upside in this attack is that the adversary can only hijack a portion of the traffic destined to an AS since it has to announce a longer prefix.

2.2. Blockchains and their potential use in BGP routing

Blockchain technology has introduced a new paradigm in distributed systems with applications spanning cryptocurrencies, smart contracts, distributed provenance, and censorship resistance [17–20]. With promising guarantees, *e.g.* immutable and append-only data management in a decentralized system, blockchains are suited to create provenance and prevent fraud in a trust-less environment.

Blockchains use various consensus protocols to arbitrate trust among networks and entities. Some of the well-known consensus protocols include PoW, PoS, PoET, PoA, and PBFT [21,22]. Each consensus protocol has its benefits and limitations which we show in Table 1. Among the five consensus protocols, we selected PoS, PoA (*Clique*), and PoET for *RouteChain* due to (1) high scalability, (2) energy efficiency,

Table 1
Comparison of five popular consensus protocols used in the blockchain systems. The table values reflect the theoretical fault tolerance in an ideal system. These values can change in a real-world system due to various factors, *e.g.*, network delay or selfish mining strategies can reduce PoW fault tolerance to ≈25% hash rate [24–26]. Since real-world system parameters can vary across different implementations, we therefore compare theoretical values in this table for a fair comparison Among all properties shown below, the throughput is measured in transactions per second (tx/second).

Properties	PoW	PoS	PBFT	*Clique*	PoET
Blockchain Type	Permisssionless	Permissionless	Permissioned	Permissioned	Permissionless
Trust Model	Untrusted	Untrusted	Semi-trusted	Semi-trusted	Untrusted
Scalability	High	High	Low	High	High
Throughput	<10	<1,000	<10,000	–	1,000
Fault Tolerance	50%	50%	33%	50%	50%
Energy Efficient	No	Yes	Yes	Yes	Yes

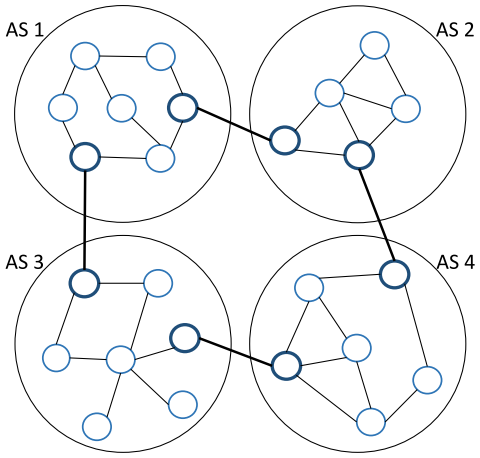


Fig. 1. An overview of ASes connected to one another. In AS 1–4, the circles inside represent routers that are responsible for intra-AS routing. Bold circles represent the border routers that are connected to other border routers through BGP protocol. To emphasize their connectivity, we color them in dark.

(3) scalability, and (4) high throughput. We excluded PoW and PBFT from *RouteChain* since they cannot be applied in the current network of 88,721 ASes [23]. PoW has a high energy consumption which can be harmful to the environment if more than 80K ASes use it for consensus over each BGP announcement. Similarly, PBFT has a high message complexity which restricts its deployment to less than 50 nodes at any time [11]. Therefore, in *RouteChain*, we only use PoS, PoET, and PoA (*Clique*) consensus protocols, and in the following, we briefly summarize them.

Clique. *Clique* is a PoA-based consensus protocol that is fast, scalable, and has a high fault tolerance. In *Clique*, a primary replica is selected to order transactions in a block and broadcast that block to the other nodes. Although the process of transaction ordering is similar to PBFT, *Clique* relies on *eventual consistency* which reduces the message complexity from $O(n^2)$, where n is the number of entities in the system to $O(1)$, thus favoring scalability. Additionally, unlike PBFT, *Clique* has a high fault tolerance and it can tolerate up to $f/2$, where f is the number of faulty replicas. In other words, *Clique* can prevent a BGP hijack as long as 50% ASes behave honestly, which is a fair and reasonable assumption to make in practice.

PoS. The PoS protocol is similar to an auction process in which each participant bids for a block. The participant with the highest bid is allowed to order transactions and release them to the rest of the network. PoS can be realized in *RouteChain* by assigning fixed tokens to each AS and the AS using those tokens order routes in the block. For correct ordering (*i.e.,* non-malicious BGP announcements), the AS can be rewarded more tokens to promote the honest behavior. As shown in Table 1, as long as no AS possesses a majority of tokens, PoS-based *RouteChain* construction will be able to prevent the BGP hijack.

PoET. PoET leverages the security properties of Trusted Execution Environment (TEE) to randomly elect a leader in the network who then orders transactions and proposes a block. PoET is currently deployed in the Hyperledger Sawtooth using Intel Software Guard Extensions (SGX) [27]. The randomized nature of leader election ensures decentralization and guarantees the blockchain consistency under the honest majority assumption.

Table 1 shows that *Clique* is used in *permissioned* network where all the network participants know each other's identities. In contrast, PoS and PoET can be applied in the *permissionless* network where participants join and leave the network at any time. However, in *RouteChain*, we enforce that the network is *permissionless* and ASes can identify each other by the AS number and the routing paths. All the ASes will be required to register their initial routes in the public blockchain ledger along with the AS number. Since *Clique* is already designed for the *permissioned* network, it can be easily applied in *RouteChain* without any significant modification to the protocol specifications. In contrast, specifications of PoS and PoET will be restricted to only honor the transactions of ASes that are already registered in the blockchain.

3. Problem statement and threat model

In the following, we describe our problem statement along with the threat model. Using this model, we build the motivation and describe the methodology of our work.

3.1. Problem statement

Broadly speaking, the problem with routing attacks on ASes involves the exploitation of a weak trust model that parameterizes the communication among ASes. Each AS has its own routing table at the gateway router, which only includes routing information of the ASes that are directly connected to it. Moreover, when a new piece of routing information is presented to the AS – that is not in conflict with its own routing policies – it readily accepts them and sets its outbound traffic to the new path. Moreover, the router at the gateway is unaware of the routing paths maintained by neighboring ASes. Lacking a global knowledge obstructs the detection of a malicious routing path when propagated. Therefore, the lack of a synchronized global knowledge along with a weak trust model lead to the routing attacks. Although those attacks are infrequent, their impact could be catastrophic, making them important to address.

Another challenge in preventing BGP hijacking is the cost associated with the existing countermeasures. For instance, one of the well-known countermeasures is the use of "Resource Public Key Infrastructure" (RPKI) [28], which enables network operators to cryptographically authorize ASes the to announce a specific prefix. To effectively use RPKI, ASes set up a route assigning authority called "Routing Origin Authorization" (ROA), that overlooks prefix authorization. While promising in theory, RPKI has some practical limitations. First, it requires all ASes to become part of a central authority and conform to its policies. In a

decentralized network of over 80,000 ASes – each with a different functional policy – achieving this agreement can be challenging. Second, this clean-slate approach towards design and deployment of routing schemes may not be acceptable to network operators.

3.2. Threat model

For BGP attacks, we assume the adversary to be a malicious AS or a group of ASes, aiming to hijack traffic of a target AS. The adversary can either launch a partial hijacking by announcing an identical prefix, or launch a complete hijacking by announcing more precise IP prefixes of the victim AS (§2.1). The adversary will exploit the weak trust model of its neighboring ASes, and expect them to change their routing paths accordingly. After announcement to the neighboring ASes, the adversary will hope its prefixes propagate swiftly through other ASes to maximize the attack severity.

Assuming a victim is a valuable AS (i.e., an AS of interest to the adversary), hosting nodes that contain sensitive or valuable information, such as Bitcoin mining pools, then a large-scale attack can be launched to eclipse the target AS. In this attack, multiple ASes can simultaneously launch a complete or partial BGP attack, thereby hindering the recovery process of the victim. In such a situation, the attack may persist for a long time, causing excessive revenue and reputation loss [29].

3.3. Motivation

The motivation of this work is to introduce a blockchain-based routing scheme to prevent BGP attacks. As identified previously, one of the major challenges in countering BGP attacks is the *weak* trust model among ASes, requiring them to trust all new prefix announcements. Blockchain can improve this trust model by augmenting consensus over prefix announcements. Blockchain, by design, incorporates consensus in a system, thereby ensuring provenance, audibility, and attribution. We intend to use the tamper-proof guarantees of blockchains to equip routing tables with an immutable ledger that tracks routing paths. Therefore, when a malicious AS announces false prefixes, the rest of the network is able to detect and discard them. Moreover, keeping in view the need of a swift consensus, we aim to design *RouteChain* in a hierarchical way to improve performance and reduce latency. At the time of writing this work, there are 88,721 ASes on the Internet [23], and using *Clique* to achieve consensus among them over a single routing path will lead to significant delays and unnecessary processing overhead. To avoid that, we intend to explore new design parameters that can be tailored to the efficiency requirements of the system. Finally, a major effort in this work has been dedicated towards the incremental deployment of *RouteChain*: our objective is to achieve a standalone system that can be integrated with the current functionalities of ASes and does not require them to modify their policies.

3.4. Methodology

In our methodology, we first set up a *RouteChain* model with a BGP announcement characterized as a blockchain transaction and a blockchain ledger that orders all those transaction. Next, we set up a blockchain testbed across various geographical locations to accurately model the block propagation delay and the consensus time among the nodes. In our testbed, we implement the rules of all three consensus protocols and enable feasible switching between those protocols without changing the overlay topology and the communication model.

We design our system in such a way that the consensus time over a new route announcement is less than the time of BGP attack. For that purpose, we take the Youtube hijacking of 2008 [30] as our baseline attack model to calculate the hijacking time. We then calculate the consensus time of each protocol in *RouteChain* and contrast it against the baseline attack timeline. As such, if ASes achieve a consensus over a malicious prefix before the prefix propagates in the network, then the ASes can reject the prefix and prevent the attack. We evaluate the performance of each consensus protocol in terms of consensus time, and the protocol with the minimum consensus time was selected as the most suitable choice for *RouteChain*.

Table 2
Symbols and Definition.

Symbols	Definition
A	ASes in world $A = a_1, a_2, \dots a_n$, where n is the total number of ASes
K	ASes subgroups $K = k_1, k_2, \dots, k_p$, where p is the total number of subgroups and $p \leq n$
$\mathcal{A}$	A group of one or more adversarial ASes
$\mathcal{V}$	Victim AS to be hijacked by $\mathcal{A}$
$\mathcal{B}_A$	Global blockchain for A
$\mathcal{T}_A$	Global blockchain consensus time for $\mathcal{B}_A$
$\mathcal{B}_K$	Subgroup blockchains, $\mathcal{B}_K = \mathcal{B}_{k_1}, \mathcal{B}_{k_2}, \dots \mathcal{B}_{k_p}$
$\mathcal{P}_A$	Global primary replica for $\mathcal{B}_A$
$\mathcal{P}_{k_i}$	Subgroup primary replica for k_i, where $1 \leq i \leq p$
$\mathcal{T}_K$	ASes consensus time in a subgroup
$\mathcal{T}_A$	Consensus time in global blockchain
$\mathcal{T}_E$	End-to-end transaction delay
$\mathcal{T}_h$	Time to hijack target AS
$\mathcal{T}_p$	communication time between two ASes.
$\mathcal{T}_v$	Blockchain verification time

4. Routechain

In this section, we present the design and analysis of *RouteChain*. In Table 2, we provide the notations that will be used throughout the rest of the paper.

4.1. System architecture

The overall design of *RouteChain* involves all ASes A sharded into K subgroups, with each subgroup sharing a single ledger. Subgroups are constructed based on their geographical proximity in order to reduce propagation delays and achieve faster consensus. Within a subgroup, each AS will maintain a subgroup ledger to keep track of routing paths that belong to all ASes within the subgroup. Having such a transparent and consistent view ensures that if any AS is targeted within the subgroup, all other ASes will be able to detect that, and take preventive measures.

Our choice of sharding ASes into groups has multiple benefits. First, maintaining a single ledger for all ASes on the Internet can have a significant storage overhead, since each AS will be forced to maintain and update the routing information of all other ASes. Second, as the blockchain size grows, the time required to validate the correctness of a new transaction increases. For every new transaction, an AS would need to check the entire blockchain to view the prior history of a path defined in the new transaction. This verification time is critical, as it contributes to the overall consensus time. Having a long single ledger would increase the verification time for each incoming transaction.

Finally, the key challenge in *RouteChain* is to obtain consensus of peers over a new route before the convergence of the legacy system. As mentioned in Section 3.3, *RouteChain* will be deployed in parallel to the legacy system, therefore our objective is to flag a bogus route before it propagates through the legacy system and leads to a hijack. In *RouteChain*, subgroups will enable a parallel processing over a new transaction. Since ASes in subgroups are associated based on their geographical (network) proximity, they can achieve faster consensus due to low propagation delays for any given transaction. As such, parallel processing within subgroups will reduce the overall consensus time, as opposed to a flat blockchain system composed of all ASes. As such, sharding ASes in subgroups is a useful for fast consensus over BGP routes.

For provenance assurance and a globally consistent view, *Route Chain* also maintains a global blockchain shared among subgroups. The

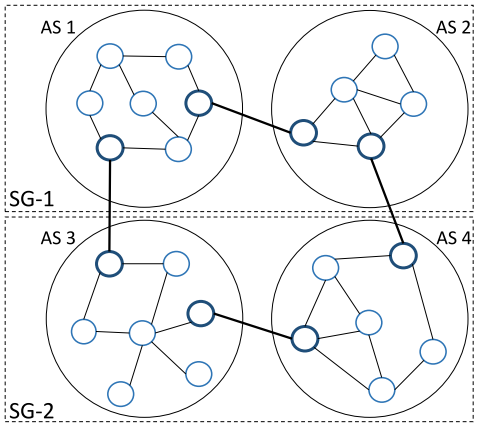


Fig. 2. An illustration of proposed architecture depicting ASes in a subgroup. SG-1 is a subgroup with two ASes, AS 1 and AS 2 in close proximity. SG-2 is another subgroup of two ASes, AS 3 and AS 4 in close proximity. Selection of subgroups can be made on any other scheme such as transit route policy.

global chain one maintains decisions on a given transaction, and does not contain detailed information of routing paths. For instance, when an AS announces his prefixes in a transaction, once the transaction gets approved by subgroups, the concerned subgroup updates its routing table while the global chain locks the transaction approval. The global chain will have all the announcements that are made by ASes in the form of a transaction. Therefore, when a new transaction is issued by an adversarial AS, the global blockchain can be consulted to track the true owner of those prefixes. The global chain however, does not include the routing paths that are maintained at the routers of an AS.

In summary, *RouteChain* is a bi-hierarchical blockchain system, consisting of a global chain shared among subgroups, and subgroup chains shared among ASes. Once a transaction is generated, it is forwarded to subgroups, and upon approval, its status is updated in the ledgers (§4.5).

4.2. Subgroup structure

A subgroup will consist of multiple ASes with a shared ledger of routing paths. The selection of ASes for a subgroup can be achieved through various design choices. For instance, grouping ASes based on their geographical proximity can reduce propagation delays. On the other hand, grouping them based on their transit relationships can have less policy conflicts. *RouteChain* is agnostic towards the policy of forming subgroups as long as they share a single ledger. In this paper, we assume the peer relationship to be driven by the incentive of geographical proximity and low delays [31]. However, as part of our future work, we will explore other mechanisms to that be adopted for better results.[1] In Fig. 2, we illustrate how four ASes from Fig. 1 can be arranged into two subgroups.

In §4.4.1, we derive optimal value of the number of subgroups K, that results in minimum delays for consensus. Since the Internet architecture is continuously evolving and the network typologies change with time, we accommodate this changing modularity in *RouteChain* by keeping the subgroup size and number of subgroups as tunable parameters. If a group of new ASes fall into the same geographical proximity, and its addition would breach the size limit (let $|k_{max}|$ be the size limit of a subgroup), then two subgroups can be extracted from the same subgroup.

[1] Since *RouteChain* is a permissioned system, therefore, when a new AS intends to join the system, it can select a subgroup based on geographical proximity and send a membership request to the subgroup primary.

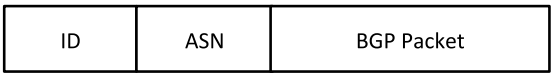


Fig. 3. Transaction data structure in *RouteChain*. The unique identifier is used to for each transaction. Unique identifier can be assigned from an incremental counter or by generating a random value. Autonomous system number (ASN) field identifies the advertising AS, and BGP packet holds the metadata of routing information.

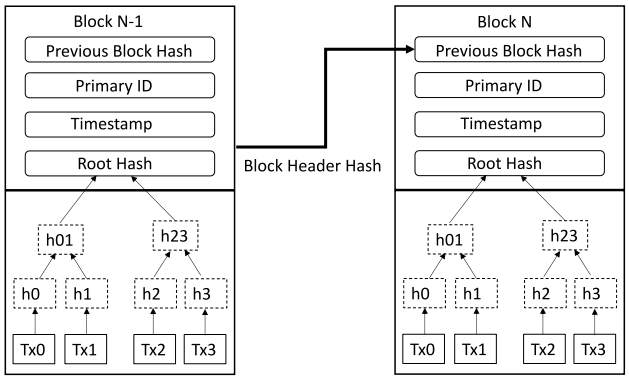


Fig. 4. Blockchain data structure for a subgroup in *RouteChain*. The block header contains previous block hash, the subgroup primary identifier, the block timestamp, and the Merkle root hash. The payload consists of transactions that are hashed in pairs to form a Merkle tree. The transaction data structure follows the specification provided in Fig. 3.

4.3. Consensus mechanism

As mentioned in §2.1, we use three consensus protocols in *Route Chain*. Based on each protocol's specifications, a primary replica P_K is selected from each subgroup K to send transactions to other replicas (ASes) [32,33]. P_K is responsible for receiving transactions, ordering transactions, computing a block, and broadcasting it to the other replicas. This process is carried out in one epoch, and at each epoch, a new primary is selected to continue the procedure.

Transaction Data Structure. In Fig. 3, we show the data structure of transaction generated by an AS. The transaction consists of a transaction ID field, an autonomous system number (ASN) field, and the BGP packet. For details on the structure of BGP packet, we refer the reader to [12] Once this transaction gets approved, it is updated in the global blockchain ledger.

Blockchain Data Structure. In Fig. 4, we show the blockchain structure for the subgroups. The block header contains (1) the previous block hash, (2) the primary replica identifier, (3) the block timestamp, and (4) the Merkle root. The block payload consists of transactions that are hashed in pairs to form a Merkle tree. The Merkle root serves as a digital fingerprint for the block header, offering protection against data tampering.

The block payload can consist of multiple transactions or a single transaction. Multiple transactions can be included if there is more than one announcement during the consensus time. For instance, consider a case where ASes start mining a new block at time t_1. The block is eventually mined at time t_2. During the consensus time $(t_2 - t_1)$, 20 new prefix announcements are received by the primary replica. Each announcement is treated as a new transaction and all transactions are mined in the subsequent block.

In case if only one announcement is received during the consensus time $(t_2 - t_1)$, or the gap between new announcements is greater than the expected consensus time (see Fig. 7 for details), only one transaction is included in the block and the Merkle root in Fig. 4 is replaced by the transaction hash.

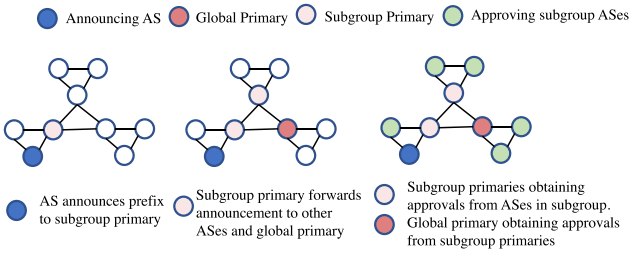


Fig. 5. An illustration showing transaction approval process. First the origin AS announces an announcement to the subgroup primary $\mathcal{P}_{K_i}$ which then forwards the transaction the global primary $\mathcal{P}_A$ and other ASes in K_i. Subsequently, $\mathcal{P}_{K_i}$ obtains approval from $|k_i|/2$ ASes and $\mathcal{P}_A$ obtains approval from $|K|/2$ subgroups. In this figure, origin AS, subgroup primaries, global primary, and approving ASes are color coded in blue, pink, red, and green, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

This blockchain data structure makes all ASes within a subgroup see all the BGP announcements. Based on import and export policies, the ASes may not want to re-advertise a path announcement. Also, an AS may do AS-PATH padding to increase the hop length of a path because it desires this path not to be used. But, our blockchain-based approach will effectively force all ASes in a subgroup to (i) re-advertise all the path announcements they received and (ii) use shortest possible paths. Although this may limit the flexibility in the import and export policies of individual ASes, it is an acceptable approach as the ASes in the same subgroup will likely trust each other and will not have an issue with sharing their prefix announcements. It is an open problem to design the blockchain data structure in a way that an AS advertising a path may choose to keep its announcement (or certain parts of it) encrypted while still adding it to blockchain for origin authorization purposes.

4.4. Analysis of RouteChain

In this section, we perform analysis of *RouteChain* in terms of design requirements and performance objectives.

4.4.1. Design analysis

In *RouteChain*, the primary replica not only sends transactions to other replicas but also awaits an approval response. Additionally, the blockchain of a subgroup gets updated only if the approved transaction concerns the routing paths of ASes within the subgroup. Otherwise, only the subgroup decision is sent back to the global primary. We take this approach for two reasons. First, a subgroup must only update its blockchain if a transaction directly concerns its routing path. Otherwise, maintaining routing paths of other subgroups is an unnecessary overhead. Second, this also reduces the size of the blockchain, making transaction verification faster.

The process of obtaining transaction approval involves: 1. An AS broadcasting transaction to its subgroup primary $\mathcal{P}_{K_i}$, 2. $\mathcal{P}_{K_i}$ forwarding transaction to global primary $\mathcal{P}_A$, 3. $\mathcal{P}_A$ broadcasting transaction to subgroups K, 4. $\mathcal{P}_{K_i}$ obtaining at least $|k_i|/2$ responses from subgroup ASes, and 5. $\mathcal{P}_A$ obtaining responses from $|K|/2$ subgroups. In Fig. 5, we provide an illustration of transaction approval process.

For consensus within a subgroup, the primary will take one propagation delay $\mathcal{T}_p$ to broadcast the transaction to all replica ASes. Each AS will take the verification time $\mathcal{T}_v$ to verify transaction from the blockchain, and $\mathcal{T}_p$ to send back the response to the primary. However, due to varying link conditions and verification capacities, the overall time of receiving $|k_i|/2$ responses might incur non-uniform delays. We characterize such delays with parameter ϵ_1. Therefore, for a given transaction, the consensus time $\mathcal{T}_K$ of a subgroup can be calculated as follows:

$$\mathcal{T}_K = \mathcal{T}_p + \mathcal{T}_v + \mathcal{T}_p + \epsilon_1 = \mathcal{T}_v + 2\mathcal{T}_p + \epsilon_1 \tag{1}$$

In the second phase of transaction confirmation, the global primary $\mathcal{P}_A$ needs $|K|/2$ responses from subgroup primaries. Similar to Eq. (1), the time to achieve this consensus would be the verification time, the propagation time, and the second overhead ϵ_2, capturing random delays for information propagation between subgroups and the global primary. Therefore, the transaction confirmation time $\mathcal{T}_A$, for the global blockchain $\mathcal{B}_A$ can be calculated as follows:

$$\mathcal{T}_A = \mathcal{T}_K + \mathcal{T}_p + \epsilon_2 = \mathcal{T}_v + 3\mathcal{T}_p + (\epsilon_1 + \epsilon_2) \tag{2}$$

In a complete transaction life cycle, the transaction will start from an AS within a subgroup and will reach the global primary through the subgroup primary. Once received, the global primary will initiate the verification process by broadcasting transactions to the subgroups. Considering the propagation delays at each step, the overall duration of a transaction $\mathcal{T}_E$ can be calculated as follows:

$$\mathcal{T}_E = 3\mathcal{T}_p + \mathcal{T}_K + \mathcal{T}_A = \mathcal{T}_v + 6\mathcal{T}_p + (\epsilon_1 + \epsilon_2) \tag{3}$$

Minimizing Delay. To defend against BGP attacks, we want the value of $\mathcal{T}_E$ to be small. From Eq. (3), it can be observed that propagation delays $\mathcal{T}_p$ linearly contribute the most towards the end-to-end duration of transaction confirmation $\mathcal{T}_E$. Therefore, in order to reduce the number of propagation delays, we need to reduce the number of messages required to process a transaction. This depends upon the number of ASes in a subgroup, and the total number of subgroups respectively. We note that in *RouteChain*, $N/2$ messages are required by the primary to commit a transaction (§4.3). This means that $\mathcal{P}_A$ requires approval from 50% replicas in both hierarchies. If we fix, K subgroups for *RouteChain*, the size of each subgroup will be A/K, and the number of approvals required will be $A/2K$, since approvals among the subgroups will happen in parallel. Next, $\mathcal{P}_A$ also require approvals from $K/2$ subgroups for the global blockchain. Therefore, for a transaction confirmation, in total, $\mathcal{P}_A$ requires $A/2K+K/2$ approvals. For simplicity, we assume that each subgroup is uniform in size such that $|k_1| = |k_2| = |k_p|$. Using that and after simplification, we obtain the following equation that shows the total number of approvals required for the transaction commitment C.

$$C = \frac{A}{2K} + \frac{K}{2} = \frac{A + K^2}{2K} \tag{4}$$

From Eq. (4), our objective is to find the suitable value of K that yields minimum value of C. Therefore, by differentiating Eq. (4) with respect to K, and setting it to 0, we obtain the optimum value of K as: $K^* = \sqrt{A}$. Note that K^* is independent of the fact that the protocol requires half of the ASes to respond. This simplifies the design and $K^* = \sqrt{A}$ becomes an optimum target as the AS count A changes. After plugging $A = 88{,}721$, the total number of ASes in the world, we get $K^* \approx 298$. This means that with number of subgroups fixed at 298, we will need minimum number of approvals for a transaction commitment. Minimum approvals will naturally have minimum propagation delays, which serves our main goal in *RouteChain*.

4.5. Security analysis

In this section, we will analyze security properties of *RouteChain* in the light of the threat model. An adversary controlling one or more ASes will try to launch a partial or complete BGP attack on a target AS. In the following, we show how *RouteChain* defends against these attacks.

Partial Attack. In a partial hijacking attack, the adversarial AS $\mathcal{A}$ announces identical BGP prefix that belongs to the victim AS $\mathcal{V}$. In the taxonomy of blockchains, this attack can be considered as double-spending, since $\mathcal{A}$ is trying to utilize a resource that is already being consumed by $\mathcal{V}$. As shown in Fig. 3, the transaction will have the same BGP packet as payload but a conflicting value in ASN field. Since the global blockchain $\mathcal{B}_A$ will have a prior transaction of same prefixes linked to the identity of $\mathcal{V}$, such an anomaly can be easily detected in *RouteChain*.

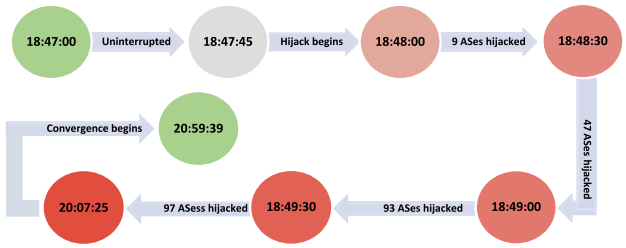


Fig. 6. Timeline of Youtube Hijacking. Notice that within one minute of the announcement, 9 ASes had changed their routes, and within 20 min 97 ASes were redirecting their traffic to AS17577.

If $\mathcal{V}$ and $\mathcal{A}$ belong to the same subgroup, the hijacking attempt will be neutralized immediately by the subgroup primary $\mathcal{P}_{K_i}$, unless $\mathcal{A}$ is itself the primary. In such a case, the transaction will be sent to the global primary $\mathcal{P}_A$. Also if $\mathcal{A}$ and $\mathcal{V}$ belong to different subgroups, the transaction will be forwarded to the $\mathcal{P}_A$ by $\mathcal{P}_{K_i}$. In both cases, the subgroup primaries will consult the global blockchain $\mathcal{B}_A$, and be able to spot the double-spent transaction. To defend against partial attacks, the response from subgroup primaries would be sufficient to obtain consensus over the transaction. As long as 50% of the subgroup primaries behave honestly, a partial hijacking can be detected and countered immediately.

Complete Attack. In a complete attack, $\mathcal{A}$ announces more specific prefixes than the ones owned by $\mathcal{V}$. As such, this behavior cannot be immediately detected as a double-spent transaction since $\mathcal{B}_A$ will not have a prior transaction linked to $\mathcal{V}$ that contains prefixes announced by $\mathcal{A}$. To detect that, *RouteChain* would require a conflict resolution from all ASes in subgroups. Each AS will consult their subgroup blockchain to see if the newly advertised routes in the transaction alters their routing paths. If true, they will observe the original path and its corresponding prefix. Next, they will locate the true owner of the prefix through the global blockchain $\mathcal{B}_A$. Accordingly, they will be able to detect that the newly announced prefix does not belong to the true owner, therefore, it is malicious. As long as 50% ASes in a subgroup and 50% of total subgroups behave honestly, the hijacking attempt can be detected and prevented in time.

Compromising 50% of ASes within a subgroup can be costly in practice. In all the well-known attacks that have been launched against ASes, only one AS or ISP has been found to be the miscreant. Therefore, in practical settings, even if the subgroup size is small, a collusion of 50% ASes is highly improbable. Furthermore, the adversary will also need support from 50% of the total number of subgroups. Therefore, for a successful attack, the adversary would require half of the total ASes in the world to be on its side. Considering the associated cost, we conclude that *RouteChain* provides high security guarantees in adversarial settings.

5. *Routechain* deployment and evaluation

In this section, we evaluate the performance of *RouteChain* by deploying a blockchain testbed that supports PoS, PoET, and *Clique*. To closely model the subgroup size, we set up 250 nodes in Docker containers and deployed them across five different cities including Francisco, New York, Singapore, London, and Frankfurt. The motivation to set up nodes across different cities was to obtain realistic values for transaction confirmation by acknowledging propagation delay over the Internet. The testbed nodes were deployed on the *digital ocean data center* [34], which is cloud service provider that hosts its data centers globally. All nodes in the data center were assigned a public IP address and the IP address mapping was specified in the Docker configuration file. The Docker containers were aggregated in servers and each server had an Intel Xeon processor with 16 GB RAM, and 500 GB hard drive.

To enable communication among the blockchain nodes, we set up a NodeJS-based communication framework that specified the overlay topology among nodes. We also installed a middleware at each container that encoded the rules of the three consensus protocol. Each node used GET and POST methods of HTTP to exchange data (*i.e.,* transactions and blocks) with other nodes. To minimize delay in the overlay communication model, we set up a completely-connected overlay topology, where each node was connected to every other node in the network.

Next, we encoded the rules of each consensus protocol in our *RouteChain* testbed. For PoS, we randomly assigned tokens between the range of 100–10,000 to each node. For simplicity, we incorporated stakes for the block within the block header. As a result, in each round, nodes ordered transactions in the block with a random stake value and broadcast the block to all the other nodes in the network. Upon receiving different blocks, a node selects the block that had the highest stakes specified in the block header.

In order to implement PoET, instead of using Intel SGX, we implemented a script with random waiting time for each node. We pursued this approach to avoid the costly implementation of Intel SGX. In practice, ASes can easily afford to set up a node with Intel SGX. In our PoET implementation, a random waiting time was allocated to each node and once the waiting time expired, the node released a new block. To avoid blockchain forks, we calibrated the waiting times using a global clock. For *Clique* implementation, we followed a round-robin scheme to select a primary replica for each round. Furthermore, in each block, we specified the next primary replica to order transactions and broadcast a block.

To summarize, we set up 250 blockchain nodes across five different cities and implemented three different consensus protocols. Our testbed represented a subgroup of ASes that approved new BGP routes proposed by ASes within the subgroup. We set up a block size of 0.5MB and calculated the end-to-end block confirmation time. Given a BGP packet size of 4,096 Bytes, a 0.5MB block size might appear to be exceedingly large containing thousands of transactions. However, as with most blockchain systems, specifying a larger block size helps in estimating the upward system scalability and respect the real-world network constraints such as congestion. We calculated the global consensus time $\mathcal{T}_A$ as the time taken by $K/2$ subgroups to return a valid response to the global primary. Since each subgroup processes transactions in parallel, $\mathcal{T}_A$ can be approximated as $2 \times \mathcal{T}_K$. Therefore, in our experiments, we compute $\mathcal{T}_K$ and derive $\mathcal{T}_A$ as $2 \times \mathcal{T}_K$.

Results. In Figs. 7 and 8, we show the average of results obtained from our testbed implementation. Fig. 7 shows the subgroup consensus time $\mathcal{T}_A$ and the global consensus time for PoS, Clique, and PoET. We found that in a subgroup of 250 ASes, PoS, Clique, and PoET confirmed the transaction in 7.6, 5, and 14.1 ms, respectively. Similarly, the global consensus time for the three protocols was 15.2, 10, and 28.2 ms, respectively. The results show that *Clique* has the minimum consensus time which makes it the most suitable protocol for *RouteChain*.

In Fig. 8, we report the maximum transaction throughput for each protocol. At the subgroup level, PoS, *Clique*, and PoET showed an achievable throughput of 129, 200, and 70 transactions per second, respectively. At the global layer, the three protocols achieved a throughput of 64.5, 100, and 35 transactions per second, respectively. Since each transaction is a new route announcement, the results show that *Clique* can globally process up to 100 BGP announcements in one second, making it highly scalable.

We now evaluate the testbed results in light of our threat model §3.2. In a partial hijack, only the consensus from the subgroup primaries is sufficient to detect the hijacking attack and neutralize it. Since the partial hijacking attack reflects a double-spending attack in the global blockchain, subgroup primaries can effectively query the blockchain and notify the global primary. Our results in Fig. 7 show that the minimum subgroup consensus time is 5 ms which can be

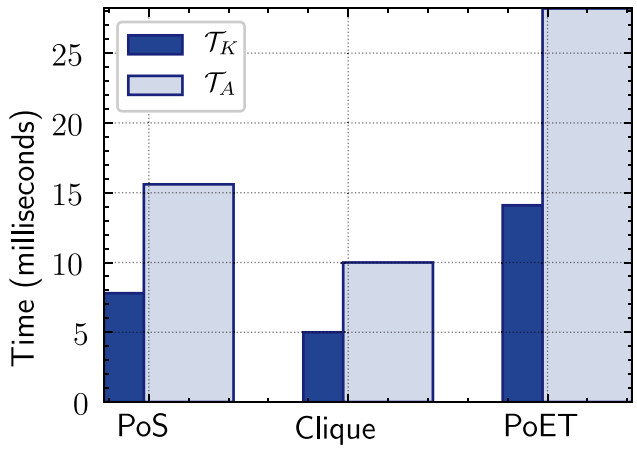


Fig. 7. The subgroup consensus time $\mathcal{T}_K$ and the global consensus time $\mathcal{T}_A$ for the three consensus protocols. Our results show that *Clique* has the minimum consensus time, making it the ideal candidate for *RouteChain*.

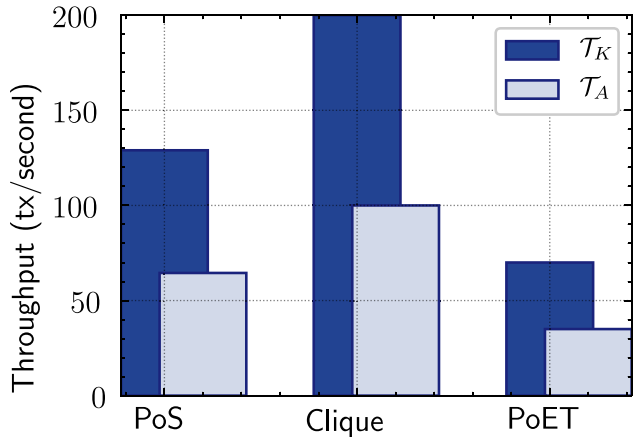


Fig. 8. Maximum transaction throughput for all the three consensus protocols at the subgroup level and the global level. Our results show that *Clique* achieves the maximum throughput.

achieved by implementing *Clique* in *RouteChain*. To prevent a complete hijacking attack, *RouteChain* acquires consensus of subgroups and ASes within a subgroup. Our results Fig. 7 show that the minimum global consensus time is 10 ms. Although the consensus time over the complete hijacking attack is higher than the partial hijacking attack, considering the timings of real world attacks (see Fig. 6), this is tolerable.

To further evaluate the effectiveness of *RouteChain*, we contrast the consensus time with the timings of Youtube's attack shown in Fig. 6. It can be observed that during the attack duration, within 1 min, 9 ASes were hijacked and within 20 min, 97 ASes were hijacked. However, our results in Fig. 7, show that all consensus protocols have a global consensus time less than 200 ms. Therefore, the BGP attack can be easily neutralized using any of the three protocols which makes *RouteChain* highly effective in practice. *RouteChain* will notify the AS operators about the malicious route before the AS adopts that route and directs its traffic .

Also note that the blockchain size in *RouteChain* is less than the blockchain size for prominent cryptocurrencies, *e.g.*, Bitcoin. A typical BGP packet size is 4096 Bytes, and in 2021, 30,000 BGP prefixes were announced everyday [35]. By multiplying the total number of announcements with the average announcement size and converting Bytes to Gigabytes, the total size of all announcements becomes 0.1144 GB per day. For an upper bound analysis, if we assume the block payload to be consisting of one BGP announcements and the block header size of 80 Bytes [36], the total increase in the blockchain size becomes 0.1167 GB $\left((4176 * 30000)/2^{30}\right)$, everyday.[2] Multiplying this value with the total number of days in a year yields a ≈42.58 GB increase in *RouteChain* size each year, which is below the average increase of the Bitcoin blockchain size (≈62 GB [38]). Given that ASes have a high bandwidth and the *RouteChain* size is optimal, we expect a tolerable storage and communication overhead of our blockchain-based implementation over the current implementations.

6. Discussion

One limitation of our work is the assignment of ASes within subgroups. To achieve ideal results, we show how subgroups can be structured to obtain consensus in minimum time (Eq. (4)), however, in practice this may not be as close to the ideal situation. We group ASes based on geographical proximity. However, in the real world [39], ASes may have conflicting interests or policies that may prevent them from being part of the same subgroup that also contains their competing ASes. However, as we mentioned earlier, geographical proximity is one policy, among others, that be used to construct *RouteChain*. As such, subgroup structure is agnostic of the underlying policy as long as it shares a single ledger.

Furthermore, a subgroup size can vary depending upon its construction policy. While obtaining consensus from 50% ASes is pertinent to the transaction verification, however, reducing their number will reduce the latency. A transaction can be processed with agreement of fewer replicas, provided that the system has a strong trust model. We view this as an avenue for future research where the performance of *RouteChain* can be evaluated based upon varying subgroup size, structure, and policy. Finally, *RouteChain* can be incrementally deployed from a small group of ASes to all ASes over the Internet. In this paper, we provide a roadmap towards secure routing through immutable route management. This can be bootstrapped at a small scale and later adapted by entities which find it useful.

7. Related work

In this section, we review notable efforts done to secure Internet routing protocols against well known attacks. Our prior work on secure BGP routing [40], explored the use of blockchain to authenticate prefix announcements. We used *Clique* consensus protocol for blockchain and conducted simulations to estimate the consensus time. In this paper, we extend the scope of prior work by analyzing three consensus protocols for *RouteChain* including Proof-of-Stake (PoS), *Clique*, and Proof-of-Elapsed Time (PoET). Additionally, we deployed a testbed of blockchain nodes for realistic evaluation of consensus time. In the following, we discuss similar works on BGP security.

Towards blockchain-based secure BGP routing, Xing et al. [41] proposed *BGPcoin*; a smart contract driven BGP framework that is implemented over Ethereum network. *BGPcoin* reduces the possibility of BGP prefix hijacking by providing secure BGP advertisements through smart contract, achieving a throughput of 5 transactions per second. Hari et al. [42] also proposed a blockchain-based secure BGP routing. They also identified caveats in using RPKI in the decentralized architecture of ASes. Although the authors did not provide simulation results, they concretely provided metrics that need to be adopted by a blockchain-based BGP solution to be practical. More precisely, they recommend the transaction throughput to be four orders of magnitude greater than Bitcoin which has a throughput of 3–7 transactions

[2] Since one transaction per block is an upper-bound estimate, having multiple transactions per block has a smaller size overhead. Moreover, there are two prominent ways to represent the data size, including the decimal representation (base 10) and the binary representation (base 2) [37] . In our work, we used the binary representation to compute the *RouteChain* size.

per second. Our experiments on *RouteChain* show that it can achieve a maximum throughput of 200 and 100 transactions per second at the subgroup layer and the global layer. Our results conform to the specifications provided in [42].

Chang et al. [43] proposed a behavioral assumption scheme, called *AS-TRUST* to design ASes reputation. *AS-TRUST* provides probabilistic trust for the ASes by evaluating their prior broadcasts and classify the outcomes based on a reputation function. Towards blockchain-based secure Domain Name Systems (DNS), Liu et al. [44] proposed a data storage method, called DecDNS that creates multiple DNS nodes in parallel to address single point-of-failure in DNS resolution.

In summary, prior works on blockchain-based BGP routing [41–43] predominantly required upgrading the method of BGP announcements, followed by their inclusion in the blockchain. As such, tailoring the process of BGP announcements mandates a foundational change in the system design which could be incompatible with the current deployments. In *RouteChain*, we do not propose any refinement to BGP announcement process. ASes can continue to operate as usual and deploy *RouteChain* as an additional security feature. This adoption will not require a fundamental change in the ecosystem, making *RouteChain* an incrementally deployable solution

There has been extensive research carried out to secure BGP without the use of blockchains. Hu et al. [45] proposed Cooperative Information Sharing Model (CoISM) to improve shortcomings of information sharing through BGP monitoring. CoISM does not modify the current processing of BGP and can be implemented to validate real BGP routes and detect fake BGP routes. Moreover, it can be deployed in various inter-domain management applications, such as intrusion detection and failure analysis of routing. Camacho et al. [46] proposed BGP eXtended Multipath (BGP-XM) for transit Autonomous Systems. BGP-XM uses algorithms including K-Best Route Optimizer to strengthen BGP multi-path routing. BGP-XM does not violate BGP functionalities when selecting routes among different ASes paths. Schlamp et al. [47] proposed a "Hijacking Event Analysis Program" *HEAP*; to filter data sources and rate validity of BGP sub-prefixes in order to defend against hijacking.

8. Conclusion

In this paper, we present a blockchain-based secure BGP routing system called *RouteChain*. *RouteChain* uses a hierarchical blockchain structure and three consensus protocols to facilitate fast and tamper-proof route management. While the blockchain ledger provides a validation source for all BGP announcements, the consensus protocols provision an agreement among ASes over the prefix nature. Combined, these two properties enable *RouteChain* to act as standalone security service that can be incrementally deployed in parallel with current operations of ASes. We develop a blockchain testbed to evaluate the effectiveness of *RouteChain* using PoS, *Clique*, and PoET consensus protocols. Our results show that *RouteChain* is capable of preventing a BGP attack in all three protocols with *Clique* being the most suitable protocol choice due to low consensus time and a high throughput.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] S. Spoto, M. Gribaudo, D. Manini, Performance evaluation of peering-agreements among autonomous systems subject to peer-to-peer traffic, Perform. Eval. 77 (2014) 1–20, http://dx.doi.org/10.1016/j.peva.2014.02.004.

[2] R. Kanzaki, S. Fujita, Peer-to-peer content delivery system with bounded traffic between autonomous systems, in: J.E. Guerrero (Ed.), International Symposium on Computing and Networking - Across Practical Development and Theoretical Research, IEEE Computer Society, Matsuyama, Japan, Dec 2013, pp. 630–632, http://dx.doi.org/10.1109/CANDAR.2013.114.

[3] V.N. Padmanabhan, L. Subramanian, An investigation of geographic mapping techniques for internet hosts, SIGCOMM Comput. Commun. Rev. 31 (4) (2001) 173–185, http://dx.doi.org/10.1145/964723.383073, URL: http://doi.acm.org/10.1145/964723.383073.

[4] P. Sermpezis, V. Kotronis, A. Dainotti, X.A. Dimitropoulos, A survey among network operators on BGP prefix hijacking, Comput. Commun. Rev. 48 (1) (2018) 64–69, http://dx.doi.org/10.1145/3211852.3211862.

[5] A. Mitseva, A. Panchenko, T. Engel, The state of affairs in BGP security: A survey of attacks and defenses, Comput. Commun. 124 (2018) 45–60, http://dx.doi.org/10.1016/j.comcom.2018.04.013.

[6] M. Jonker, A. Pras, A. Dainotti, A. Sperotto, A first joint look at DoS attacks and BGP blackholing in the wild, in: Internet Measurement Conference IMC, ACM, Boston, USA, Nov 2018, pp. 457–463, http://dx.doi.org/10.1145/3278532.3278571, URL: http://doi.acm.org/10.1145/3278532.3278571.

[7] J. Qiu, L. Gao, S. Ranjan, A. Nucci, Detecting bogus BGP route information: Going beyond prefix hijacking, in: International Conference on Security and Privacy in Communication Networks SecureComm Nice, France, IEEE, Sept 2007, pp. 381–390, http://dx.doi.org/10.1109/SECCOM.2007.4550358.

[8] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, G. Danezis, Consensus in the age of blockchains, 2017, arXiv preprint arXiv:1711.03936.

[9] S.D. Angelis, Assessing security and performances of consensus algorithms for permissioned blockchains, 2018, CoRR arXiv:1805.03490.

[10] M. Saad, J. Spaulding, L. Njilla, C.A. Kamhoua, S. Shetty, D. Nyang, D.A. Mohaisen, Exploring the attack surface of blockchain: A comprehensive survey, IEEE Commun. Surv. Tutor. 22 (3) (2020) 1977–2008, http://dx.doi.org/10.1109/COMST.2020.2975999.

[11] A. Ahmad, M. Saad, J. Kim, D. Nyang, D. Mohaisen, Performance evaluation of consensus protocols in blockchain-based audit systems, in: 2021 International Conference on Information Networking, ICOIN, IEEE, 2021, pp. 654–656.

[12] Y. Rekhter, T. Li, S. Hares, A border gateway protocol 4 (BGP-4), RFC 4271 (2006) 1–104.

[13] B.R. Smith, J. Garcia-Luna-Aceves, Securing the border gateway routing protocol, in: In Proceeding of the Global Telecommunications Conference, GLOBECOM., IEEE, 1996, pp. 81–85.

[14] Developer, BGP hijacking overview. Routing incidents prevention and defense mechanisms., 2018, https://bit.ly/2E2wB4H.

[15] E. Biersack, Q. Jacquemart, F. Fischer, J. Fuchs, O. Thonnard, G. Theodoridis, D. Tzovaras, P. Vervier, Visual analytics for BGP monitoring and prefix hijacking identification, IEEE Netw. 26 (6) (2012) 33–39, http://dx.doi.org/10.1109/MNET.2012.6375891.

[16] E.L. Wong, V. Shmatikov, Get off my prefix! the need for dynamic, gerontocratic policies in inter-domain routing, in: IEEE/IFIP International Conference on Dependable Systems and Networks, DSN, IEEE Compute Society, Hong Kong, China, June 2011, pp. 233–244, http://dx.doi.org/10.1109/DSN.2011.5958222.

[17] R. Neisse, G. Steri, I. Nai-Fovino, A blockchain-based approach for data accountability and provenance tracking, in: Proceedings of the 12th International Conference on Availability, Reliability and Security, ACM, 2017, http://dx.doi.org/10.1145/3098954.3098958, https://dl.acm.org/citation.cfm?doid=3098954.3098958.

[18] S. Omohundro, Cryptocurrencies, smart contracts, and artificial intelligence, AI Matters 1 (2) (2014) 19–21, http://dx.doi.org/10.1145/2685328.2685334, http://doi.acm.org/10.1145/2685328.2685334.

[19] A.E. Azzaoui, P.K. Sharma, J.H. Park, Blockchain-based delegated quantum cloud architecture for medical big data security, J. Netw. Comput. Appl. 198 (2022) 103304, http://dx.doi.org/10.1016/j.jnca.2021.103304.

[20] A. Ahmad, M. Saad, A. Mohaisen, Secure and transparent audit logs with *BlockAudit*, J. Netw. Comput. Appl. 145 (2019) http://dx.doi.org/10.1016/j.jnca.2019.102406.

[21] M. Saad, A. Mohaisen, Towards characterizing blockchain-based cryptocurrencies for highly-accurate predictions, in: IEEE Conference on Computer Communications Workshops, INFOCOM Workshops, Honolulu, HI, USA, April 2018, pp. 704–709, http://dx.doi.org/10.1109/INFCOMW.2018.8406859.

[22] M. Saad, L. Njilla, C.A. Kamhoua, A. Mohaisen, Countering selfish mining in blockchains, CoRR (2018) arXiv:1811.09943.

[23] RIR, Autonomous systems in the world, in: Regional Internet Registries Statistics - RIR Delegations - World - Autonomous System Number Statistics - Sorted By Number, 2018, https://tinyurl.com/yaz73jnb.

[24] C. Decker, R. Wattenhofer, Information propagation in the bitcoin network, in: International Conference on Peer-To-Peer Computing, IEEE P2P, Trento, Italy, 2013, pp. 1–10, http://dx.doi.org/10.1109/P2P.2013.6688704.

[25] I. Eyal, E.G. Sirer, Majority is not enough: bitcoin mining is vulnerable, Commun. ACM 61 (7) (2018) 95–102, http://dx.doi.org/10.1145/3212998.

[26] H. Guo, X. Yu, A survey on blockchain technology and its security, Blockchain: Res. Appl. 3 (2) (2022) 100067, http://dx.doi.org/10.1016/j.bcra.2022.100067, URL: https://www.sciencedirect.com/science/article/pii/S2096720922000070.

[27] Sawtooth, Hyperledger sawtooth, 2019, URL: https://sawtooth.hyperledger.org/docs/core/releases/latest/architecture/poet.html,

[28] R. Bush, Clarifications to BGP origin validation based on resource public key infrastructure (RPKI), RFC 8481 (2018) 1–5, http://dx.doi.org/10.17487/RFC8481.

[29] P. Bangera, S. Gorinsky, Impact of prefix hijacking on payments of providers, in: International Conference on Communication Systems and Networks, COMSNETS, Bangalore, India, Jan 2011, pp. 1–10, http://dx.doi.org/10.1109/COMSNETS.2011.5716486.

[30] S. Goldberg, Why is it taking so long to secure internet routing? Commun. ACM 57 (10) (2014) 56–63, http://dx.doi.org/10.1145/2659899, URL: http://doi.acm.org/10.1145/2659899.

[31] J. Hawkinson, T. Bates, Guidelines for creation, selection, and registration of an autonomous system (AS), RFC 1930 (1996) 1–10, http://dx.doi.org/10.17487/RFC1930.

[32] E.R. Weippl, S. Katzenbeisser, C. Kruegel, A.C. Myers, S. Halevi (Eds.), Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, vienna, Austria, october 24-28, 2016, ACM, 2016, URL: http://dl.acm.org/citation.cfm?id=2976749.

[33] Y. Sompolinsky, A. Zohar, Accelerating bitcoin's transaction processing. Fast money grows on trees, not chains, IACR Cryptol. EPrint Arch. 2013 (2013) 881, URL: http://eprint.iacr.org/2013/881.

[34] The developer cloud, DigitalOcean URL: https://www.digitalocean.com/.

[35] G. Huston, BGP in 2021 - BGP UPDATES, APNIC Blog, 2022, URL: https://blog.apnic.net/2022/01/13/bgp-updates-2021/.

[36] B. Community, Mastering bitcoin, O'Reilly Online Learning, O'Reilly Media, Inc., URL: https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch07.html.

[37] Community, Gigabytes conversion, Data Units Conversion URL: https://www.gbmb.org/gigabytes.

[38] B. Community, Bitcoin Blockchain Size, URL: https://ycharts.com/indicators/bitcoin_blockchain_size#:~:text=Basic.

[39] W. Liang, J. Bi, Y. Xia, C. Hu, RPIM: inferring BGP routing policies in ISP networks, in: Global Communications Conference, GLOBECOM, Houston, Texas, USA, IEEE, Dec 2011, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2011.6133970.

[40] M. Saad, A. Anwar, A. Ahmad, H. Alasmary, M. Yuksel, A. Mohaisen, RouteChain: towards blockchain-based secure and efficient BGP routing, in: 2019 IEEE International Conference on Blockchain and Cryptocurrency, ICBC, IEEE, 2019, pp. 210–218.

[41] Q. Xing, B. Wang, X. Wang, BGPcoin: Blockchain-based internet number resource authority and BGP security solution, Symmetry 10 (9) (2018) 408.

[42] A. Hari, T.V. Lakshman, The internet blockchain: A distributed, tamper-resistant transaction framework for the internet, in: ACM Workshop on Hot Topics in Networks, HotNets '16, ACM, New York, NY, USA, 2016, pp. 204–210, http://dx.doi.org/10.1145/3005745.3005771, URL: http://doi.acm.org/10.1145/3005745.3005771.

[43] J. Chang, K.K. Venkatasubramanian, A.G. West, S. Kannan, B.T. Loo, O. Sokolsky, I. Lee, AS-TRUST: a trust quantification scheme for autonomous systems in BGP, in: International Conference on Trust and Trustworthy Computing, TRUST, 2011, pp. 262–276.

[44] J. Liu, B. Li, L. Chen, M. Hou, F. Xiang, P. Wang, A data storage method based on blockchain for decentralization DNS, in: In Proceeding of the Third IEEE International Conference on Data Science in Cyberspacem, DSC, 2018, pp. 189–196.

[45] N. Hu, B. Wang, X. Liu, Cooperative monitoring BGP among autonomous systems, Secur. Commun. Netw. 8 (10) (2015) 1943–1957.

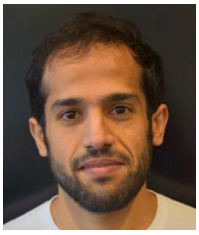
[46] J.M. Camacho, A. García-Martínez, M. Bagnulo, F. Valera, BGP-XM: BGP eXtended multipath for transit autonomous systems, Comput. Netw. 57 (4) (2013) 954–975.

[47] J. Schlamp, R. Holz, Q. Jacquemart, G. Carle, E.W. Biersack, HEAP: reliable assessment of BGP hijacking attacks, IEEE J. Sel. Areas Commun. 34 (6) (2016).

Muhammad Saad completed his Ph.D. in Computer Science at the University of Central Florida. At UCF, Saad was a member of the Security Analytics and Research Lab (SEAL) advised by Prof. David Mohaisen. During his PhD., he worked on the security of distributed systems with an emphasis on the blockchain attack surface.

Afsah Anwar completed his Ph.D. in Computer Science at the University of Central Florida. He obtained his Bachelors of Technology degree in Electronics and Communication Engineering from the Jamia Millia Islamia (JMI), in New Delhi, India. At UCF, he was a member of the Security and Analytics Lab (SEAL), advised by Prof. David Mohaisen. Afsah software security.

Ashar Ahmed completed his PhD. in Computer Science at the University of Central r Science at Ghulam Ishaq Khan University in 1999. At UCF, he was advised by Prof. David Mohaisen. Ashar worked on blockchains and distributed systems.



Hisham Alasmary is an Assistant Professor at King Khalid University. He obtained his Ph.D. from the Department of Computer Science at the University of Central Florida, and his M.Sc. degree in Computer Science from The George Washington University, in Washington, D.C., USA, in 2016. At UCF, he was a member of the Security and Analytics Lab (SEAL), where he worked on computer and network systems security.

Murat Yuksel (Senior Member, IEEE) received the Ph.D. degree in computer science from Rensselaer Polytechnic Institute, Troy, NY, USA in 2002. He is currently a Professor with the ECE Department, University of Central Florida (UCF), Orlando, FL, USA. Prior to UCF, he was with the CSE Department, University of Nevada Reno, Reno, NV, USA as a Faculty Member until 2016. His research interests are in the area of networked, wireless, and computer systems.

David Mohaisen earned his M.Sc. and Ph.D. degrees from the University of Minnesota in 2011 and 2012, respectively. He is currently an Associate Professor at the University of Central Florida, where he directs the Security and Analytics Lab (SEAL). Before joining UCF, he held several posts, in academia and industry: as an Assistant Professor at the University at Buffalo, (Senior) Research Scientist at Verisign Labs, and a Member of the Engineering Staff at the Electronics and Telecommunication Research Institute (ETRI). His research interests fall in the broad areas of networked systems and their security, adversarial machine learning, IoT security, AI security, and blockchain security. Among other services, he is currently an Associate Editor of IEEE Transactions on Mobile Computing and IEEE Transactions on Parallel and Distributed Systems. He is a senior member of ACM (2018) and IEEE (2015).