RouteChain: Towards Blockchain-based Secure and Efficient BGP Routing

Muhammad Saad
University of Central Florida
saad.ucf@knights.ucf.edu

Afsah Anwar
University of Central Florida
afsahanwar@Knights.ucf.edu

Ashar Ahmad
University of Central Florida
ashar@cs.ucf.edu

Hisham Alasmary
University of Central Florida
hisham@Knights.ucf.edu

Murat Yuksel
University of Central Florida
Murat.Yuksel@ucf.edu

Aziz Mohaisen
University of Central Florida
mohaisen@cs.ucf.edu

***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 results in an attack surface which is exploited by malicious entities to hijacking 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 validate strengths of our design by simulations and show its effectiveness by drawing a case study with the Youtube hijacking of 2008. Our proposed scheme is a standalone service that can be incrementally deployed without the need of a central authority.**

***Index Terms*—Blockchain; Autonomous Systems; BGP**

I. 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].

978-1-7281-1328-9/19/$31.00 2019 IEEE

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.

Embracing the policy-based functional challenges as well as the security risks in routing, we propose a new scheme called *RouteChain*, which counters BGP hijacking by leveraging design constructs of 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 trust 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 in order 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 to be exchanged among peers (ASes). We use a Proof-of-Authority (PoA)-based consensus protocol called *Clique* [8] to create consensus among ASes, and securely lock the routing information in a mutually shared ledger. We perform

a comprehensive analysis to establish the feasibility of our proposed scheme and validate it by extensive simulations. 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 known BGP attack on Youtube, with the consensus time in *RouteChain*. Our results show that *RouteChain* neutralizes the attacker's efforts by developing swift consensus among honest ASes concerning routing states.

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 system called *RouteChain* that prevents BGP hijacking. 2) We provide analytical evaluation of our design and validate its achievable outcomes through simulations. 3) We show the feasibility of *RouteChain* by drawing a case study from the well-known Youtube's BGP hijacking case, where *RouteChain* is shown to neutralize the attack. 4) Our proposed solution is shown to be incrementally deployable; easily implemented alongside existing protocols.

Organization. The rest of the paper is organized as follow. In §II, we provide the background and preliminaries of our work. In §III, we introduce the problem statement, the threat model, the motivation, and the methodology. In §IV, we present our proposed solution *RouteChain*, along with its design and analysis. In §V, we present the simulations and results. In §VI, we discuss the advantages and limitations of our work. That is followed by related work and conclusion in §VII and §VIII.

II. Background and Preliminaries

In this section, we briefly review the background of ASes, BGP, and blockchains, aligned with the scope of this work.

A. 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 networks it is hosting, 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 two neighboring ASes can be peering or customer-provider engagement, and typically means that a level of trust exists 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 [9]. 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 [9], [10]. In this paper, we are concerned with eBGP that pivots communication among edge routers of neighboring ASes. Figure 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.

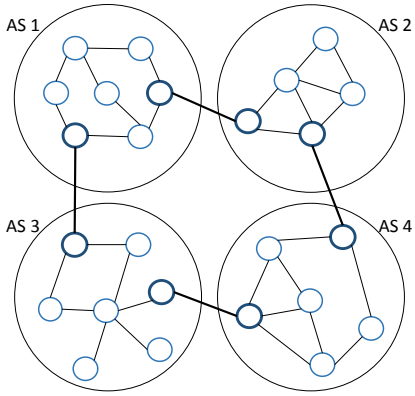

Fig. 1. An overview of ASes connected to one another. In AS 1–4, the circles inside represent routers that are responsible for intra-AS routing. Bold circles represent the border routers that are connected to other border routers through BGP protocol. To emphasize their connectivity, we color them in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

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 [11]. 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 [12], [13]. 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.

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. Therefore, 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 [11].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.

B. Blockchains

Blockchain technology has introduced a new paradigm in distributed systems with applications spanning cryptocurrencies, smart contracts, distributed provenance, and censorship

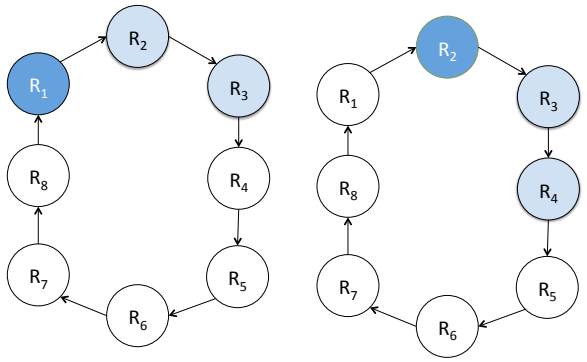

Fig. 2. Clique uses a round-robin scheme to select primary node, and two secondary nodes, the primary is responsible for proposing a new block. In case of failure of a primary node the block is proposed by the secondary nodes. Note: R1 is the primary and R2 and R3 and secondary for the first epoch, for the second epoch R2 is the primary and R3 and R4 and the secondary.

resistance [14], [15], [16], [17]. With promising guarantees including immutable and append-only data management in a decentralized system, blockchains are well 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 the proof-of-work (PoW), proof-of-stake (PoS), proof-of-elapsed time (PoET), proof-of-authority (PoA), and the practical Byzantine fault tolerance (PBFT) [18], [19]. Each scheme has its merits and caveats. However, best suited to the requirements of our design, we use a modified form of PoA, known as *Clique*. For more details on consensus schemes and their uses, we refer the reader to [20].

Clique. *Clique* is a PoA-based consensus scheme that is fast, scalable, and has high fault tolerance. In *Clique*, a primary is chosen to order transactions and broadcast to the nodes for each round of block execution. In contrast to PBFT, where block execution takes four phases with $O(n^2)$ message complexity, a block is executed in a single phase with $O(1)$ message complexity in *Clique*. For each epoch, a primary replica is selected in a round-robin, and the view of the primary is replicated on the mutually shared ledger. Moreover, *Clique* has a high fault tolerance and can endure up to $f/2$ faulty replicas. In Figure 2, we show how a primary replica is selected in *Clique*, and later in §IV-C we present the design implementation of *Clique* in our system.

III. 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.

A. 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 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) [21], 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.

B. 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 (§II-A). 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 [22].

C. Motivation

The motivation of this work is to introduce a blockchain-based routing scheme to prevent BGP attacks. 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

TABLE I
SYMBOLS AND DEFINITION

Symbols	Definition
A	ASes in world $A = a_1, a_2, ...a_n$
K	ASes subgroups $K = k_1, k_2, ..., k_p$, where $p \leq n$
$\mathcal{A}$	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, $\mathcal{B}_K \mathcal{B}_{k_1}, \mathcal{B}_{k_2}, ...\mathcal{B}_{k_p}$
$\mathcal{P}_A$	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

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 *Modus Operandi* of ASes and does not require them to modify their policies.

D. Methodology

First we explore possible ways in which *RouteChain* can be structured to meet the requirements of *Clique*. We design our system in such a way that the consensus time among ASes is less than the time of hijacking. To that end, we set Youtube hijacking of 2008 [24] as our baseline attack model to derive the hijacking time. Next, we compare the consensus time in *RouteChain* with the hijacking time obtained from the baseline attack model to analyze the strength of our design in adversarial settings. We do that by carrying event driven simulations that mimic the real-world conditions of the Internet. Since *RouteChain* is a modular structure, we tailor its design to achieve consensus over routes in minimum time.

IV. *RouteChain*

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

A. 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 each 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, since 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 §III-C, *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, *RouteChain* also maintains a global blockchain shared among subgroups. The 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 receiving approvals, its status is updated in the ledgers (§IV-E).

B. 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 each subgroup shares a single ledger. In this paper, we assume the peer relationship to be driven by the incentive of geographical proximity and low delays [25]. However, as part of our future work, we will explore other mechanisms to that be adopted for better results. In Figure 3, we illustrate how four ASes from Figure 1 can be arranged into two subgroups.

In §IV-D1, 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, therefore we accommodate this changing modularity in *RouteChain* by keeping the subgroup size and number of subgroups flexible. For instance,

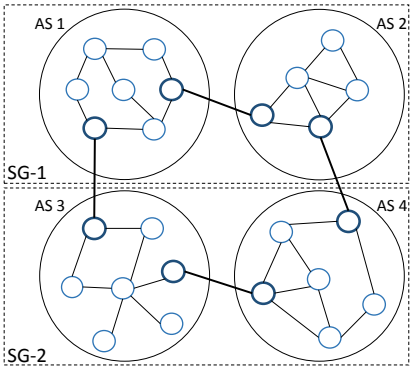

Fig. 3. An overview of subgroup ASes. SG-1 is a subgroup with two ASes, AS 1 and AS 2 in close proximity. SG-2 is another subgroup of two ASes, AS 3 and AS 4 in close proximity. Selection of subgroups can be made on any other scheme such as transit route policy.

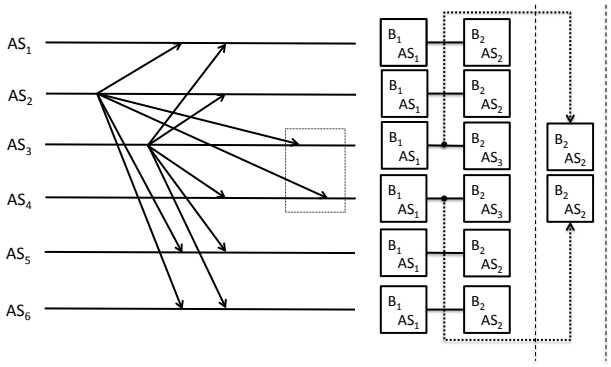

Fig. 4. Fork in clique consensus protocol, AS_2 is the primary replica who broadcasts a block. Due to delays and latency AS_4 does not get the block on time. Instead it receives the block published by the next primary AS_3. This leads to a fork. The fork can be resolved easily by mapping the prior known identity of the primary with the expected block.

let $|k_{max}|$ be the upper size limit of a subgroup. If a group of new ASes falls into the same geographical proximity, and its addition would breach the size limit, then two subgroups can be extracted from the same subgroup.

C. Consensus Mechanism

As mentioned in §II-B, in *RouteChain*, we use *Clique* as the consensus protocol. For each subgroup in K, a primary replica $\mathcal{P}_K$ is selected to send transactions to other replicas (ASes) as shown in Figure 4. $\mathcal{P}_K$ 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. The selection of a primary follows a round-robin scheme. Therefore, in a subgroup size of $|k_i|$, if an AS becomes the primary, it has to wait for $|k_i| - 1$ epochs to become the primary again.

In *Clique*, it is possible that a block broadcasted by a primary gets does not reach all replicas in time. This can happen due to network delays or protocol malfunction. Moreover, it is also possible that in such non-deterministic network behavior, the block of the next primary arrives at a replica before the block of the current primary. This may lead to a disordered sequence of blocks or a fork, as shown in Figure 4.

Fig. 5. Transaction data structure in *RouteChain*. The unique identifier is used to for each transaction. Autonomous system number (ASN) field identifies the advertising AS, and BGP packet holds the metadata of routing information.

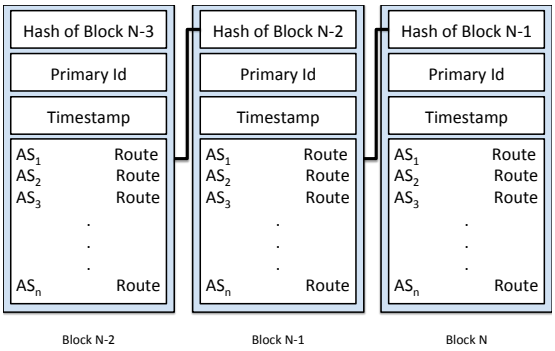

Fig. 6. Blockchain data structure for a subgroup in *RouteChain*. The block contains a timestamp, hash of the previous block, and block payload. Block payload includes autonomous systems and their routes. All ASes within a subgroup have a transparent view of routes that belong to each AS.

Resolving a fork in *Clique* is straightforward. Since the identity of the primary replica is known, therefore for each epoch, all ASes expect the new block with the identity of the primary replica, as shown in Figure 6. Therefore, when blocks are received out of the order, replicas can resolve the forks and align their view of the blockchain. We illustrate this aspect of fork formation and resolution in Figure 4.

Transaction Data Structure. In Figure 5, 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 [9] Once this transaction gets approved, it is updated in the global blockchain ledger.

Blockchain Data Structure. In Figure 6, we provide the structure of the blockchain for each subgroup. The blockchain consists of a timestamp that shows the time of block publication. The block also has a unique primary Id, and the hash of a previous block. In the block payload, we have all the ASes along with their complete routing paths. When a new block is computed, the routing paths are updated based on the newly approved transaction.

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.

D. Analysis of RouteChain

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

1) Design Analysis: In a *Clique* blockchain consisting of N replicas, a transaction requires approval from $N/2$ replicas. In *RouteChain*, we make a minor adjustments to *Clique* such that the primary replica not only sends transactions to other replicas but also awaits an approval response. In the conventional design of *Clique*, the primary only broadcasts a transaction or a block and all other replicas accept the payload. However, suitable to our design, we expect other replicas to send back their approval for the transaction.

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. Firstly, 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. Secondly, 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 the global primary $\mathcal{P}_A$, 3) $\mathcal{P}_A$ broadcasting transaction to all subgroups K, 4) $\mathcal{P}_{K_i}$ obtaining at least $|k_i|/2$ responses from subgroup replicas, 5) and $\mathcal{P}_A$ obtaining responses from $|K|/2$ subgroups

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 to obtain $|K|/2$ responses from subgroup primaries. Similar to (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 through will reach the global primary through the subgroup primary. Once received, the the global primary will initiate the verification process by broadcasting transactions to all subgroups. Taking into account propagation delays at each step, the end-to-end 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 (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 know that in *Clique*, $N/2$ messages are required by the primary to commit a transaction (§IV-C). 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 $\mathcal{C}$.

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

From (4), our objective is to find the suitable value of K that yields minimum value of $\mathcal{C}$. Therefore, by differentiating (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 Clique 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*.

E. 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 Figure 5, 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*.

If $\mathcal{V}$ and $\mathcal{A}$ belong to the same subgroup, the hijacking attempt will be neutralized immediately by the subgroup

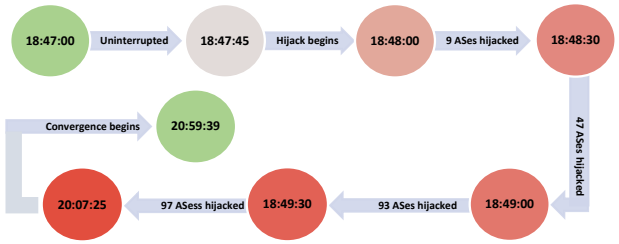

Fig. 7. Timeline of Youtube Hijacking. Notice that within one minute of the announcement, 9 ASes had changed their routes, and within 20 minutes 97 ASes were redirecting their traffic to AS17577.

primary $\mathcal{P}_{K_i}$, unless $\mathcal{A}$ is itself the primary. In such a case, the transaction will be sent to the 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.

V. Simulation and Results

In this section, we present simulations and experiments performed to validate our theoretical analysis. In particular, we focus on achieving quick consensus over a malicious route announcement. Since *RouteChain* will be deployed in parallel with the existing infrastructure, it will not mitigate the attack completely. Instead, within few seconds, we expect *RouteChain* to notify all ASes about the validity of the announced prefix. Once notified, ASes can discard the announcement to prevent further propagation, and curtail

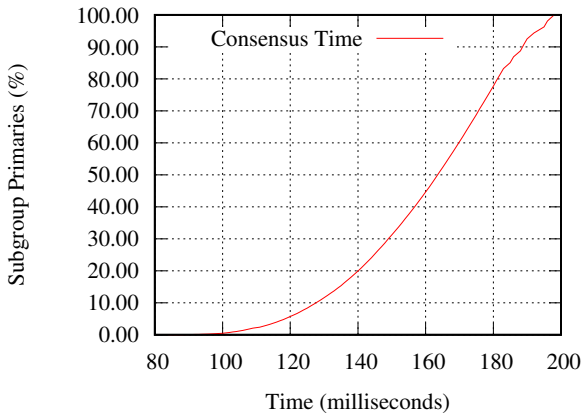

Fig. 8. Consensus time of subgroups during a partial hijacking. Notice that the network is able to detect the malicious broadcast within 200 milliseconds.

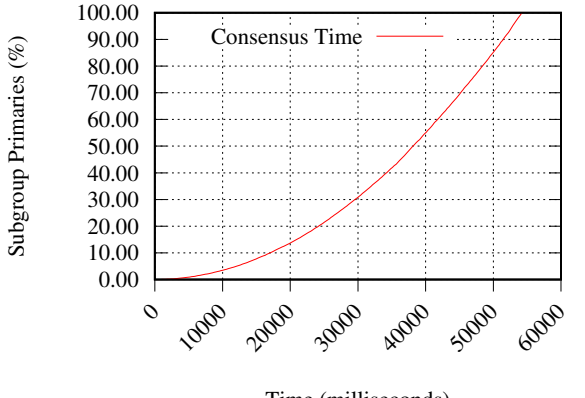

Fig. 9. Consensus time of subgroups during a complete hijacking. Subgroups and their ASes are able to detect the attack within 54.23 seconds.

damages. Towards that, we draw parallels with the *YouTube* Hijacking of 2008, and show how *RouteChain* is able to provide effective countermeasures.

Based on the design principle to reduce the time and message complexity, we consider the following parameters in our design: (1) the subgroup size A/K, (2) the number of subgroups forming the global blockchain K, (3) the subgroup consensus time $\mathcal{T}_K$, (4) the global blockchain consensus time $\mathcal{T}_E$, and (5) propagation and verification delays.

We formulate simulations based on (1)–(3), and we take practical values of AS propagation delays and blockchain verification time from the prior work [26], [27], [28]. Using these values, we set parameters in our simulator and model subgroup consensus and global consensus. Consensus time in a subgroup is the round-trip time from the transaction broadcast till the approval of $|k_i|/2$ replicas. For simulations, we record the time taken to receive a confirmation from each AS. Therefore, the consensus time becomes the difference between the start time of transaction broadcast and the time taken to receive acknowledgement from the last expected AS. Similarly global consensus time is the time taken to receive the acknowledgment from the slowest subgroup.

Next, we analyze the effectiveness of *RouteChain*, in the light of our threat model §III-B. In a partial hijack, only the consensus from the subgroup primaries is sufficient to detect the hijacking attack and neutralize it. Since the par-

tial hijacking attack reflects a double-spending attack in the global blockchain, subgroup primaries can effectively query the blockchain and notify the global primary. To that end, we simulate this scenario and show the consensus among subgroups over a malicious route in Figure 8. It can be observed that for a partial hijacking attack, the consensus is achieved within 200 milliseconds.

To prevent a complete hijacking attack, *RouteChain* acquires consensus of subgroups and ASes within a subgroup. We simulate that by recording the round-trip time between subgroup primaries and their ASes, and the round-trip time between the global primary and subgroup primaries. We plot our simulation results in Figure 8. It can be noticed that a complete hijacking attack can be detected within 54.24 seconds. Although, the consensus time over the complete hijacking attack is higher than the partial hijacking attack, considering the timings of real world attacks, this is tolerable.

To further evaluate the effectiveness of *RouteChain*, we contrast the consensus time with the timings of Youtube's attack. In Figure 7, we provide the timeline of the attack. It can be observed that during the attack duration, within 1 minute, 9 ASes were hijacked and within 20 minutes, 97 ASes were hijacked. Therefore, considering the short consensus time of *RouteChain*, we can assert with confidence that our system will notify the ASes about the attack while it is in its initial stages. Network operators can configure their rotuers to switch back to old routes once an attack is detected.

VI. Discussion

As shown by simulation, *RouteChain* is able to expose the BGP announcements to all ASes in 54.24 seconds. More specifically, for a partial hijacking, as outlined in Figure 7, *RouteChain* is able to obtain consensus from all ASes within 200 milliseconds. If *RouteChain* was deployed by ASes during the Youtube hijacking, it would have been able to prevent the hijacking when only 4 ASes were compromised. This could have saved traffic and revenue loss for Youtube AS.

Since *RouteChain* is a standalone service that runs in parallel with the operations of an AS, it cannot completely prevent all ASes from attack. Consensus among ASes is a function of time ((1)–(3)), and as such, transaction propagation and verification among all ASes may take some time. As a result, the threat of a hijacking cannot be completely eliminated even with *RouteChain*, however, as shown by simulations, the damage prevention is significant.

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 ((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 [29], 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.

VII. Related Work

In this section, we review notable efforts done to secure Internet routing protocols against well known attacks. Towards blockchain-based secure BGP routing Xing *et al.* [30] 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 enabled authentication. Hari *et al.* [31] also proposed a blockchain-based secure BGP routing. They also identified caveats in using RPKI in the decentralized architecture of ASes. However, their proposal did not include a design blueprint that can be effectively used among ASes. Chang *et al.* [32] 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.* [33] proposed a data storage method, called DecDNS that creates multiple DNS nodes in parallel to address single point-of-failure in DNS resolution.

There has been extensive research carried out to secure BGP without the use of blockchains. Hu *et al.* [34] 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.* [35] 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.* [36] proposed a "Hijacking Event Analysis Program" *HEAP*; to filter data sources and rate validity of BGP sub-prefixes in order to defend against hijacking.

VIII. Conclusion

In this paper we present a blockchain-based secure BGP routing system called *RouteChain*. *RouteChain* uses a bi-hierarchical blockchain structure and *Clique* consensus protocol to facilitate fast and tamper-proof route management. While the blockchain ledger provides a validation source for all prefixes, *Clique* enables swift consensus among ASes over the nature prefix broadcast. 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 validate achievable objectives of *RouteChain* through discreet-event simulations, and our results show that *RouteChain* can effectively curtail a BGP attack while it is in its initial stage.

Acknowledgement:. 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.

REFERENCES

[1] S. Spoto, M. Gribaudo, and D. Manini, "Performance evaluation of peering-agreements among autonomous systems subject to peer-to-peer traffic," *Perform. Eval.*, vol. 77, pp. 1–20, 2014. [Online]. Available: https://doi.org/10.1016/j.peva.2014.02.004

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

[3] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 173–185, Aug. 2001. [Online]. Available: http://doi.acm.org/10.1145/964723.383073

[4] P. Sermpezis, V. Kotronis, A. Dainotti, and X. A. Dimitropoulos, "A survey among network operators on BGP prefix hijacking," *Computer Communication Review*, vol. 48, no. 1, pp. 64–69, 2018. [Online]. Available: https://doi.org/10.1145/3211852.3211862

[5] A. Mitseva, A. Panchenko, and T. Engel, "The state of affairs in BGP security: A survey of attacks and defenses," *Computer Communications*, vol. 124, pp. 45–60, 2018. [Online]. Available: https://doi.org/10.1016/j.comcom.2018.04.013

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

[7] J. Qiu, L. Gao, S. Ranjan, and 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. [Online]. Available: https://doi.org/10.1109/SECCOM.2007.4550358

[8] S. D. Angelis, "Assessing security and performances of consensus algorithms for permissioned blockchains," *CoRR*, vol. abs/1805.03490, 2018. [Online]. Available: http://arxiv.org/abs/1805.03490

[9] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (BGP-4)," *RFC*, vol. 4271, pp. 1–104, 2006.

[10] B. R. Smith and J. Garcia-Luna-Aceves, "Securing the border gateway routing protocol," in *In proceeding of the Global Telecommunications Conference, GLOBECOM*. IEEE, 1996, pp. 81–85.

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

[12] E. Biersack, Q. Jacquemart, F. Fischer, J. Fuchs, O. Thonnard, G. Theodoridis, D. Tzovaras, and P. Vervier, "Visual analytics for BGP monitoring and prefix hijacking identification," *IEEE Network*, vol. 26, no. 6, pp. 33–39, 2012. [Online]. Available: https://doi.org/10.1109/MNET.2012.6375891

[13] E. L. Wong and 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, Hong Kong, China*. IEEE Compute Society, June 2011, pp. 233–244. [Online]. Available: https://doi.org/10.1109/DSN.2011.5958222

[14] R. Neisse, G. Steri, and 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, https://dl.acm.org/citation.cfm?doid=3098954.3098958.

[15] S. Omohundro, "Cryptocurrencies, smart contracts, and artificial intelligence," *AI Matters*, vol. 1, no. 2, pp. 19–21, Dec. 2014, http://doi.acm.org/10.1145/2685328.2685334.

[16] G. Governatori, F. Idelberger, Z. Milosevic, R. Riveret, G. Sartor, and X. Xu, "On legal contracts, imperative and declarative smart contracts, and blockchain systems," *Artif. Intell. Law*, vol. 26, no. 4, pp. 377–409, 2018. [Online]. Available: https://doi.org/10.1007/s10506-018-9223-3

[17] A. Ahmad, M. Saad, M. Bassiouni, and A. Mohaisen, "Towards blockchain-driven, secure and transparent audit logs," 2018.

[18] M. Saad and 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. [Online]. Available: https://doi.org/10.1109/INFCOMW.2018.8406859

[19] M. Saad, L. Njilla, C. A. Kamhoua, and A. Mohaisen, "Countering selfish mining in blockchains," *CoRR*, vol. abs/1811.09943, 2018. [Online]. Available: http://arxiv.org/abs/1811.09943

[20] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "SoK: Consensus in the Age of Blockchains," 2017, https://arxiv.org/abs/1711.03936.

[21] R. Bush, "Clarifications to BGP origin validation based on resource public key infrastructure (RPKI)," *RFC*, vol. 8481, pp. 1–5, 2018. [Online]. Available: https://doi.org/10.17487/RFC8481

[22] P. Bangera and 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. [Online]. Available: https://doi.org/10.1109/COMSNETS.2011.5716486

[23] RIR, "Autonomous systems in the world," 2018, https://tinyurl.com/yaz73jnb.

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

[25] J. Hawkinson and T. Bates, "Guidelines for creation, selection, and registration of an autonomous system (AS)," *RFC*, vol. 1930, pp. 1–10, 1996. [Online]. Available: https://doi.org/10.17487/RFC1930

[26] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 293–306, 2001. [Online]. Available: https://doi.org/10.1109/90.929852

[27] Y. Takahashi, K. Eguchi, A. Itoh, and K. Ishii, "Analysis of propagation-delays in high-speed bipolar gates," in *International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS, Nusa Dua Bali, Indonesia*. IEEE, Nov 2015, pp. 327–330. [Online]. Available: https://doi.org/10.1109/ISPACS.2015.7432790

[28] J. E. Pazmiño and C. Rodrigues, "Simply dividing a bitcoin network node may reduce transaction verification time," *The SIJ Transactions on Computer Networks & Communication Engineering (CNCE)*, vol. 3, no. 2, pp. 17–21, 2015.

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

[30] 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.

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

[32] J. Chang, K. K. Venkatasubramanian, A. G. West, S. Kannan, B. T. Loo, O. Sokolsky, and 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.

[33] J. Liu, B. Li, L. Chen, M. Hou, F. Xiang, and 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.

[34] N. Hu, B. Wang, and X. Liu, "Cooperative monitoring BGP among autonomous systems," *Security and Communication Networks*, vol. 8, no. 10, pp. 1943–1957, 2015.

[35] J. M. Camacho, A. García-Martínez, M. Bagnulo, and F. Valera, "BGP-XM: BGP extended multipath for transit autonomous systems," *Computer Networks*, vol. 57, no. 4, pp. 954–975, 2013.

[36] J. Schlamp, R. Holz, Q. Jacquemart, G. Carle, and E. W. Biersack, "HEAP: reliable assessment of BGP hijacking attacks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, 2016.