Global Distributed Secure Mapping of Network Addresses

Supraja Sridhara
François Wirz
ETH Zurich
Switzerland

Joeri de Ruiter
Caspar Schutijser
SIDN Labs
The Netherlands

Markus Legner
Adrian Perrig
ETH Zurich
Switzerland

Abstract

Next-generation Internet architectures are being designed and deployed to overcome limitations of today's Internet. One such architecture with an increasing production deployment is SCION [23], which also includes a transition mechanism to support an incremental deployment and coexistence with the legacy IP-based Internet: the SCION–IP gateway. This mechanism—and similar mechanisms in other next-generation architectures—requires a distributed system to translate between old (IP) and new (SCION) addresses at an Internet scale and must connect the different public-key infrastructures to enable secure operation.

In this paper, we describe such a system for the SCION architecture. A gossip protocol distributes mappings between legacy IP and SCION addresses throughout the SCION network, and SCION's control-plane PKI and the Resource Public Key Infrastructure (RPKI) protect the authenticity of the individual mappings. We provide a prototype implementation and demonstrate that it scales to today's Internet with approximately one million IP prefixes.

CCS Concepts

• **Networks** → **Naming and addressing**; **Security protocols**; *Public Internet*; Denial-of-service attacks; • **Security and privacy** → **Security protocols**; Denial-of-service attacks.

Keywords

network security, PKI, RPKI, distributed system, accountability, address mapping, gossip protocols, responsible Internet

ACM Reference Format:
Supraja Sridhara, François Wirz, Joeri de Ruiter, Caspar Schutijser, Markus Legner, and Adrian Perrig. 2021. Global Distributed Secure Mapping of Network Addresses. In *ACM SIGCOMM 2021 Workshop on Technologies, Applications, and Uses of a Responsible Internet (TAURIN '21), August 23, 2021, Virtual Event, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3472951.3473503

1 Introduction

Over the past two decades, the Internet has permeated many aspects of our society, and has evolved into of the world's most critical infrastructures. Unfortunately, the protocols at the core of today's Internet were not designed with its current scale, applications, and threat landscape in mind. This is shown by regular reports of security incidents—e.g., where traffic is accidentally rerouted [10, 26, 30] or maliciously hijacked [7, 18]. To address ever new security issues, we need a responsible Internet with controllability, accountability, and transparency as its design goals [13].

An approach to meet these goals is to fundamentally redesign core Internet protocols. SCION is a next-generation Internet architecture that achieves this—with a particular focus on security, dependability, sovereignty, and availability [23]. SCION provides end hosts with more control and transparency over their traffic through path-aware networking, where the sender can select the path their traffic should take among a set of paths offered by the network [31]. End hosts can select paths based on, for example, which networks they trust, jurisdictions to avoid, or properties like latency or bandwidth. Accountability is provided through the use of digital signatures on all control-plane messages.

Switching to a new Internet architecture has been considered highly improbable at best and impossible at worst. Besides network operators having to upgrade networking equipment, applications also would need to be rewritten to make use of the new protocols and functionality. Therefore, even if the new architecture offers substantially better service, a long period of partial deployment of the new system and coexistence with today's Internet must be expected. To enable legacy applications to start using SCION, the SCION–IP gateway (SIG) was introduced [23, §10.3], which allows to establish a tunnel between two IP networks through an intermediate SCION network. To provide this service, the SIG at the ingress point of the SCION network must be able to determine an appropriate egress SIG based on the destination IP address. While, for small networks, these mappings can be configured manually, this is no longer feasible as the SCION network continues to grow.

In this paper, we introduce the SCION–IP address-mapping system (SIAM), a system to translate between legacy IP and SCION addresses and transfer the authorizations in the Resource Public Key Infrastructure (RPKI) to SCION for use with the SIG. This way we build a secure bridge between the current Internet and a new Internet architecture, allowing users to already make use of the benefits of SCION in a state of partial deployment and without having to completely switch over all their applications and network devices. Adhering to the design principles of a responsible Internet, our design allows ASes to control when their packets are routed through SCION by publicly (de-)registering mappings; the use of digital signatures provides accountability, ensures that only authorized entities can register mappings, and enables other ASes to verify if their mappings have been registered correctly.

SIAM does not depend on any functionality that is specific to SCION. Therefore it can also be applied to other architectures with a public-key infrastructure (PKI) where a mapping needs to be made between legacy IP addresses and the new architecture's addresses.

TAURIN '21, August 23, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8639-5/21/08…$15.00
https://doi.org/10.1145/3472951.3473503

2 Background

RPKI. Currently, IP prefixes and autonomous-system (AS) numbers are assigned to network operators by the regional Internet registry (RIR) of their respective geographical region. Originally there was no authoritative source that could be used to determine whether a particular AS was authorized to originate a particular IP prefix. As a result, simple mistakes or malicious actions could lead to ASes announcing another party's IP prefixes and so receive traffic that was intended for the other party [7, 10, 18, 26, 30].

RPKI [16] was proposed to prevent such *hijacking* attacks by cryptographically binding resources, such as AS numbers and IP prefixes, to network operators using certificates. RPKI's trust anchors are self-signed certificates of the five RIRs; based on these, they can hand out resources by signing certificates for particular resources—e.g., an IP prefix. The owner of the key pair that belongs to the certificate can then distribute the resources further or authorize their use. To authorize an AS to announce a particular IP prefix, operators can publish *route origin authorizations* (ROAs). A ROA is a statement that is signed with the private key belonging to the certificate of the relevant IP prefix. This way only the owner of the IP prefix can authorize an AS to announce the prefix.

SCION and the SCION–IP Gateway. To improve routing scalability and network sovereignty, SCION groups ASes into *isolation domains* (ISDs), each of which is administered by a small subset of ($\lesssim$10) ASes called *core ASes*; other ASes are *non-core ASes*. SCION's *control-plane PKI* (CP-PKI), the counterpart to RPKI in today's Internet, fundamentally incorporates this ISD structure: Each ISD defines its own *trust-root configuration*, specifying the root keys for the CP-PKI, which are then used to create AS certificates within the CP-PKI. SCION also introduces an *anycast* system for control-plane services, in which packets with a destination set to certain *service addresses* in a specific AS are forwarded by the destination AS to the corresponding service if it exists within that AS.

Already today, there exists a global SCION deployment in the SCIONLab testbed [15, 22] and an intercontinental SCION-based production network [1, 29]. To simplify the transition to a SCION Internet, entities who own a legacy AS number obtain the same AS number within the SCION network [25].

In addition, SCION introduces the *SCION–IP gateway* (SIG) [23, §10.3] to improve incremental deployability and interoperability with the legacy IP Internet. The SIG's main application is to enable legacy IP end systems and applications to communicate through a SCION network and benefit from its properties. When a SIG receives an IP packet, it consults its local database to determine if it has a mapping of the destination address to a remote SCION AS with a SIG service. If such a mapping exists, it encapsulates the IP packet into a SCION packet and sends it via the SCION network to the remote SIG, where the packet is decapsulated and the original IP packet is forwarded to the destination. Besides routing packets through the SIG, legacy network devices and end hosts do not need to be aware of either the SIG or the SCION network.

3 Problem Description

A crucial requirement for the correct operation of the SIGs is the mapping from IP addresses to SCION ASes. To use SIGs as a transition mechanism in an Internet-scale network with tens of thousands of ASes [3], an automatic configuration system is required. This paper describes how such a distributed SCION–IP address-mapping system (SIAM) can be constructed in a secure and efficient way.

Concretely, SIAM should provide the following functionalities:

F1 An entity that controls a SCION AS and is authorized to originate a legacy IP prefix can publicly register a mapping between this IP prefix and the SCION AS, thus indicating that the IP prefix is reachable through the SCION network.

F2 These mappings should be dynamic at timescales of several hours—i.e., they can be updated and removed at any time by the owner of an IP prefix.

F3 A SIG can query SIAM to either obtain the SCION AS for a particular IP prefix or a statement that no mapping exists.

To provide the desired security properties we require SIAM to

S1 prevent hijacking attacks—i.e., an AS should not be able to add mappings for IP prefixes which it is not authorized to announce;

S2 prevent flooding attacks on SIGs—i.e., a SIG should only be able to create mappings to itself;

S3 be resilient to downgrade attacks to the legacy Internet where SCION connections are possible; and

S4 ensure high availability—i.e., the system should not have components that represent single points of failure.

Scale. We design SIAM to work at Internet scale. According to the CIDR report [3], there are currently fewer than 1 million IP prefixes announced in the Internet with ~10 prefixes per AS on average. The number of SCION ISDs is difficult to estimate; as one suggestion is that they be formed by countries [23, §3.5], we expect their number to be smaller than ~1000. As SIAM is a *transition mechanism* and thus only required during partial deployment, we conclude that designing SIAM for up to 1000 ISDs with a total of up to 10^6 IP–AS mappings is sufficient.

Attacker Model. In our design and analysis, we consider a Dolev–Yao attacker [9] who controls the network and can inject, drop, reroute, and modify packets, but cannot break cryptographic primitives. For S4 we must also assume that packets eventually arrive at the correct destination. The attacker controls a subset of SCION and legacy ASes and a (limited) number of SIAM components.

4 Strawman Approaches

Before describing our system with a gossip protocol at its heart, we convey why we do not use other seemingly suitable approaches.

Blockchains are obvious candidates for our distributed database of mappings between IP and SCION addresses: there are multiple writers that do not necessarily trust each other, and there is no obvious universally trusted central authority. However, a more detailed analysis shows that most of the core features of blockchains are irrelevant for our problem [12] and the additional processing and storage overhead is unnecessary: While multiple entities modify the same database, they work on non-overlapping subsets of entries as the allowed entries are defined externally through RPKI and the SCION CP-PKI and do not depend on the current or past state of the database. In particular, different entries in the database are independent from each other and can be verified based solely on the signatures with keys from RPKI and the SCION CP-PKI, which means that there is no need to preserve the history of mappings.

Merkle trees are a related approach to build append-only logs as used, for example, in Certificate Transparency [11]. However, these logs are not primarily intended to be used for data lookup, which is our main goal, but to detect misbehavior of trusted entities within the system. In particular, the *append-only* property is not needed for our problem setting. In our setting, most entities have limited authority—defined by RPKI and the SCION CP-PKI—and are typically fully trusted within their regions of authority. The few cases where entities *can* misbehave and cause issues, in particular related to property S3, can be solved with simpler systems.

5 SIAM System Design

SIAM enables an entity that controls both a legacy AS and a SCION AS with a SIG (with the same number, see §2) to publish a mapping from an IP prefix it owns to the SCION AS. Such a mapping (F1) enables the AS to receive IP traffic through the SCION network. To improve scalability, SIAM introduces intermediate *mapping services* (MSes) located in SCION core ASes to mediate between SIGs and the globally distributed *publishing infrastructure* (PI).

SIAM can be extended to support SCION-only AS numbers. For this we could make use of a proposal in the IETF for *Resource Tagged Attestations* [21], which describes how key material in RPKI can be used to sign arbitrary data. For example, in our case this could be used to certify that the owner of an IP prefix authorizes a specific SCION AS (which does not need to match the one in a ROA) to announce its prefix. To support this case it is particularly important to use the SCION CP-PKI in addition to RPKI to ensure property S2.

5.1 Components

The system deploys components that form the global PI as well as components local to a SCION ISD for submitting and retrieving lists of mappings to/from the PI. An overview of all SIAM components and their interaction is shown in Fig. 1 on the following page.

Publishing Gossip Node (PGN). The PGNs are nodes in the PI that accept lists containing all mappings of ASes within an ISD and respond to queries for specific lists. PGNs are globally deployed in core ASes and form a gossip network that achieves eventual consistency. For SIAM, eventual consistency is acceptable as entries can be independently verified based on attached signatures and RPKI, and PGNs provide signed statements of non-existence of missing lists. PGNs propagate the lists in their local store in periodic propagation intervals ($t_{\text{prop-pgn}}$).

Publishing List Node (PLN). To discover other PGNs, both PGNs and MSes (see below) rely on a secondary gossip network of PLNs, which are deployed in all core ASes with either a PGN or an MS. A PGN registers with a PLN by sending its ISD and AS number (IA), which the PLN stores in a local list. PLNs periodically propagate the lists in their local store to other PLNs. As PLNs are not configured with addresses of nearby PLNs, they rely on SCION's anycast feature (see §2) to discover other PLNs using small ping packets.

Mapping Service (MS). To use SIAM, an ISD must deploy an MS (and, consequently, a PLN) in at least one of its core ASes. The MS accepts mappings from SIGs within the same ISD, validates them based on both the CP-PKI and RPKI, combines them to form a list, and submits the list to the PI. Periodically, it pulls lists of all other ISDs from the PI, validates the mappings, and stores them locally in the form $ipPrefix \longrightarrow ia$ ($ia = isd : as$). Based on the stored data, it responds to queries for IP addresses from SIGs in the same ISD.

SCION–IP Gateway (SIG). A SIG creates mappings $map_{ip,as}$ for the prefixes that it is authorized to announce, signs them with its SCION CP-PKI key, and submits them to an MS in its ISD. To be able to correctly encapsulate IP packets, SIGs query the MS for IP-to-SCION-AS mappings, which they then store locally.

5.2 Overview

We summarize the end-to-end interaction of the components using the example network in Fig. 1. In this example, AS A in ISD 1 ($ia = 1:\mathtt{A}$) is authorized to use the IP prefix 192.0.2/24. The SIG $sig_{2:\mathtt{D}}$ in AS 2:D receives a legacy IP packet with destination address 192.0.2.1. SIAM enables $sig_{2:\mathtt{D}}$ to fetch a mapping for this IP address and tunnel the packet over the SCION network to AS 1:A.

AS 1:A deploys a SIG $sig_{1:\mathtt{A}}$, which registers the prefix with SIAM by sending $map_{\mathtt{192.0.2/24},\mathtt{A}}$ to the MS in the core AS B, $ms_{1:\mathtt{B}}$. The MS $ms_{1:\mathtt{B}}$ validates the mapping and adds it to a list $list_1$, which contains all mappings for ISD 1.

The MS $ms_{1:\mathtt{B}}$ sends this list to $pgn_{3:\mathtt{X}}$. The PGNs form a gossip network (see §5.1), and $pgn_{3:\mathtt{X}}$ sends $list_1$ in its gossip message to the other PGNs. Eventually, $list_1$ reaches $pgn_{4:\mathtt{Y}}$. When pulling lists from this PGN, the MS in AS 2:C obtains $list_1$; it validates $map_{\mathtt{192.0.2/24},\mathtt{A}}$ in $list_1$, and locally stores (192.0.2/24 → 1:A).

When $sig_{2:\mathtt{D}}$ receives a legacy IP packet with destination address 192.0.2.1, it needs the SCION AS number of the host with IP 192.0.2.1. It first consults its local mapping storage. If there is no rule that matches this destination address, it queries the MS $ms_{2:\mathtt{C}}$ for the mapping. The MS $ms_{2:\mathtt{C}}$ looks up its local store for the mapping and performs longest-prefix matching, which matches the prefix 192.0.2/24. The MS replies to the SIG with {192.0.2.1, 192.0.2/24, 1:A}. The SIG $sig_{2:\mathtt{D}}$ adds this to its local mapping storage and can now tunnel the IP packet through the SCION network with 2:A as SCION destination AS.

5.3 Messages and Protection

In this section we discuss the messages exchanged between the SIAM components described in §5.1 and illustrated in Fig. 1. The message formats are shown in Table 1; parameters of SIAM components and messages and their default values are listed in Table 2.

PGN Startup. As discussed in §5.1, PGNs are global components in SIAM that are deployed in core ASes. $\boxed{\text{0a}}$ When a PGN is started, it registers itself by sending its IA (pgn_{ia}) to the PLN it was configured with. The PLN that receives the registration checks the CP-PKI signatures on the message to ensure that the register request is from the correct AS. The PLN then adds the IA to its local store.

PLN Gossip. $\boxed{\text{0b}}$ PLNs periodically, in time intervals $t_{\text{ping-pln}}$, send out pings using SCION's anycast feature to SCION core ASes up to h_{pln} hops away. The ping interval $t_{\text{ping-pln}}$ and number of hops h_{pln} are configured per PLN at startup. The pings are used for detecting liveness of other PLNs and for announcing presence when a new PLN is added to the gossip network. $\boxed{\text{0c}}$ PLNs periodically combine all the pgn_{ia}s in their local store into $list_{pgn_i}$ and propagate it to k_{pln} PLNs chosen from the PLNs discovered using $\boxed{\text{0b}}$. The propagation interval $t_{\text{prop-pln}}$ and the number of PLNs k_{pln} are configured per PLN at startup.

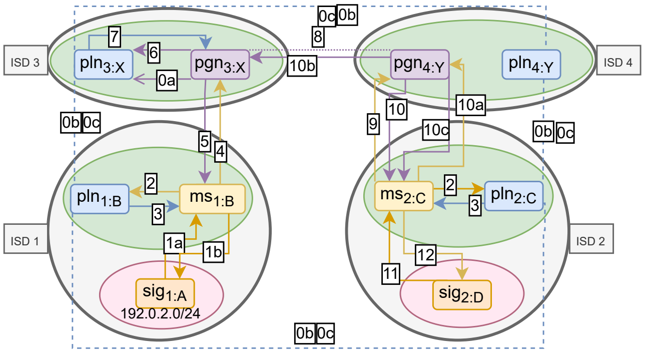

Figure 1: Overview of the SIAM system in an example SCION network with four ISDs. The green and pink ellipses denote core and non-core ASes, respectively. The numbered interactions are described in §5.3 and Table 1.

Table 1: SIAM message formats. All messages additionally contain the sender's signature based on the SCION CP-PKI, which remains attached to the message.

No.	From	To	Description	Message Format
0a	PGN	PLN	register with PLN	pgn_{ia}
0b	PLN	PLN	PLN ping	pln_{ia}
0c	PLN	PLN	PLN gossip	$list_{pgn_{ia}} = [pgn_{ia}]$
1a	SIG	MS	register an IP–AS mapping	$map_{ip,as} = \{ip, as, \text{add/remove}, ts_{\text{create}}\}$
1b	MS	SIG	confirmation token	$\text{Hash}(map_{ip,as}), ts_{ms-sig}$
4	MS	PGN	send mapping list	$obj_{ms} = \{list_{isd} = [map_{ip,as}], ts_{ms}\}$
5	PGN	MS	confirmation token	$\text{Hash}(obj_{ms}), ts_{pgn}$
8	PGN	PGN	PGN gossip	$[obj_{ms}], empty_{isd} = [\text{Sign}_{pgn-ia}\{\text{“empty''}, ts_{\text{create}}, isd\}]$
10	PGN	MS	response to request for mapping lists	$[obj_{ms}], [empty_{isd}]$
12	MS	SIG	mapping from IP address to AS	$ipAddr, ipPrefix, ia$
12a	MS	PGN	request ISD list	$isd, f_{\text{ms}}, t_{\text{empty-ms}}$

Table 2: SIAM parameters and their default values chosen as a tradeoff between dynamic operation and efficiency.

Parameter	Value	Comp.	Description
k_{pln}	8	PLN	Number of PLNs to propagate to
$t_{\text{ping-pln}}$	10 min	PLN	PLN ping interval
$t_{\text{prop-pln}}$	10 min	PLN	PLN list-propagation interval
h_{pln}	1–3	PLN	Hops for PLN ping
$t_{\text{push-ms}}$	1 h	MS	Mapping lists push interval
$validity_{\text{ms}}$	2 d	MS	Validity period for mapping lists
$t_{\text{prop-pgn}}$	10 min	PGN	PGN list-propagation interval
k_{pgn}	8	PGN	Number of PGNs to propagate to
$t_{\text{pull-ms}}$	1 h	MS	Mapping lists pull interval
$t_{\text{empty-ms}}$	1 d	MS	Validity period for empty token
f_{ms}	10	MS	Fault parameter

Register Mapping with SIAM. $\boxed{1a}$ A SIG deployed in AS *as* creates $map_{ip,as}$ for a prefix *ip* that it is authorized to originate as shown in Table 1 and sends it to an MS in a core AS of its ISD. The action add/remove can be used to specify if the AS wants to register/de-register a mapping and the ts_{create} ensures that consecutive add/remove messages can be ordered. The MS validates the origin of the message using the CP-PKI signature and validates that *as* is authorized to originate *ip*. To validate the mapping the MS uses the standard RPKI validation approach where it contacts RPKI repositories which use ROAs published by RIRs to ascertain the validity of a mapping. If the mapping is valid and the action is "add", it is stored locally along with the CP-PKI signature. If the action is "remove", the older "add" object is removed from the local store. $\boxed{1b}$ The MS acknowledges the mapping registration by sending back a hash of $map_{ip,as}$ along with a timestamp $t_{\text{ms-sig}} = t_{\text{curr}}$. The SIGs can use this token as proof against misbehaving MSes if a mapping that was submitted was not added to the PI.

Send Mappings to PI. The MS periodically, in time intervals $t_{\text{push-ms}}$, combines all the mappings in its local store into a list ($list_{isd} = [map_{ip,as}]$) and sends it to the PI. $\boxed{2}$ If it does not have a locally configured or cached PGN to which to send the list, the MS sends a request to the PLN it was configured with. $\boxed{3}$ The PLN replies to the MS with $list_{pgn_{ia}}$ from its local store. $\boxed{4}$ The MS picks a PGN to send the list of mappings to based on its local policy. It forms a message ($obj_{ms} = \{list_{isd}, ts_{ms}\}$) and sends it to the PGN. The timestamp ts_{ms} defines until when the list is valid and is calculated as $ts_{ms} = t_{\text{curr}} + validity_{\text{ms}}$ where t_{curr} is the current time and $validity_{\text{ms}}$ is configured per MS at startup.

The PGN verifies the CP-PKI signature on the message to validate its origin, checks that the ISD of the sender and the $list_{isd}$ are the same, and locally stores the signed list. $\boxed{5}$ The PGN replies with a signed message containing a hash of the MS's message, $\text{Hash}(obj_{ms})$, and a timestamp, $ts_{pgn} = t_{\text{curr}}$. This confirmation token can be used by the MS to prove misbehavior of PGNs.

If the mapping lists have not changed, MSes could send signed hashes of $list_{isd}$ with an updated expiration time instead of the full obj_{ms} to reduce communication overhead; PGNs would then send the signed hashes in their gossip messages.

PGN Gossip. The PGN periodically, in time intervals $t_{\text{prop-pgn}}$, propagates gossip messages to a number of PGNs in different ISDs configured by k_{pgn} at startup. The gossip messages contain two objects: a list of mappings ($[obj_{ms}]$) and a list of signed "empty tokens" used to prevent downgrade attacks, see §6. For the list of mappings, it combines all fresh (i.e., $ts_{ms} \geq t_{\text{curr}}$) mapping lists obj_{ms} in its local store. The PGN contacts the AS for a list of all known ISDs [23, §5.2]. It then iterates over the ISDs for which a mapping list (obj_{ms}) does not exist in its local store, and creates tokens $empty_{isd} = \text{Sign}_{pgn-ia}(\text{“empty''}, ts_{\text{create}}, isd)$, signed with the CP-PKI, where $ts_{\text{create}} = t_{\text{curr}}$. The PGN also includes all the $empty_{isd}$ that it has received.

$\boxed{6}$ To find PGNs to send the list to, the PGN contacts the PLN it was configured with at startup. $\boxed{7}$ The PLN replies with the $list(pgn_{ia})$ from its local store. $\boxed{8}$ The PGN picks k_{pgn} based on its local policy to send the list to. The PGN forms the message as defined in Table 1 and sends it to the selected PGNs.

A PGN that receives the gossip message validates the CP-PKI signatures of each of the obj_{ms} in the list, checks that the object is fresh, and locally stores obj_{ms}. The PGN iterates over the empty tokens, checks in its local store if a mapping list for the ISD with the empty token exists, and, if not, stores the empty tokens indexed by ISD number. It then replies to the sender with a list of mappings $[obj_{ms}]$ from its local store that are newer than or not present in the received gossip message and signed empty objects for ISDs without mapping lists. This process corresponds to a bidirectional exchange of the current state. The newly added entries are included in future gossip messages of the PGN as long as the object is fresh.

Pull Mapping Lists from PI. $\boxed{9}$ The MS, in time intervals $t_{\text{pull-ms}}$, queries the PGN for lists of mappings. To find a PGN, it contacts the PLN it was configured with as explained in §5.3 and picks a PGN based on its local policy (e.g., a PGN in the same ISD or an ISD it trusts). $\boxed{10}$ The PGN that receives the request combines all mapping lists (obj_{ms}) in its local store that are not stale and the signed empty tokens ($empty_{isd}$) and sends them to the MS.

Validate and Reverse Mapping Lists. The MS that receives the list ($[obj_{ms}]$) from the PGN performs two checks on each of the obj_{ms}:

- Validate the CP-PKI signature on obj_{ms}.
- Check that the entry is not stale, i.e. $ts_{ms} \geq t_{\text{curr}}$.

Afterwards, the MS validates each entry $map_{ip,as}$ in the mapping list ($list_{isd}$) by checking the CP-PKI signature and using RPKI to check for a valid ROA authorizing *as* to originate *ip*. If the validation succeeds, the MS locally stores the *as* indexed by *ip*, which allows for fast lookup when SIGs request the AS for a particular IP address.

The MS iterates over the empty tokens list ($[empty_{isd}]$) it received and checks if they are fresh, that is $t_{\text{curr}} - ts_{\text{create}} \leq t_{\text{empty-ms}}$. It then stores the tokens in its local store indexed by ISD number. To ensure that it has full information of the network, the MS checks that for each ISD in the network it either received a valid list or has f_{ms} valid empty tokens from different ISDs in its store. The "fault parameter" f_{ms} is configured at MS startup and defines the number of faulty PGN nodes the MS will tolerate.

Fetch Missing Lists. $\boxed{10a}$ If, for a particular ISD, both checks fail, the MS queries the list for the ISD by sending a message to a PGN as shown in Table 1. The PGN that receives this message checks in its local store if a fresh list for the *isd* is present. If there is no fresh list, it checks in its local store for at least f_{ms} empty tokens for *isd* with $t_{\text{curr}} - ts_{\text{create}} \leq t_{\text{empty-ms}}$ and signed by different ISDs. $\boxed{10b}$ If f_{ms} valid tokens are not present, it contacts the required number of PGNs in different ISDs for the *isd* list. The PGNs that receive this request check their local store for the *isd* list and either send back the corresponding obj_{ms} or the signed empty token $empty_{isd}$. The PGN $\text{pgn}_{4:\text{Y}}$ waits to receive an obj_{ms} with the ISD list $list_{isd}$ or f_{ms} signed empty objects whichever occurs first. The PGN can limit the f_{ms} to prevent DoS attacks by MSes. $\boxed{10c}$ It then replies to the requesting MS with either a list of signed empty objects or the actual ISD list. Through this process the MS can ascertain that it has all the mappings that are published in SIAM.

Request IP-to-AS Mapping. If a SIG receives an IP packet with a destination for which it has no mapping stored locally, it uses SIAM to lookup the SCION AS number corresponding to the IP address. $\boxed{11}$ The SIG sends a request with the IP address to an MS in any of the core ASes in its ISD. $\boxed{12}$ The MS looks up its local store for IP prefixes that match the IP address. If there is more than one mapping in the MS local store, the MS performs a longest-prefix match for the IP address and returns a message as shown in Table 1 to the SIG. If there are no matching prefixes the MS replies to the SIG with with an empty *ia* field. Instead of querying the MS on demand, the SIG can also regularly fetch the mapping state.

6 Security Analysis

In the following, we explain how SIAM achieves the properties laid out in §3 and defends against common attack scenarios.

Hijacking and Flooding Attacks. To achieve properties S1 and S2, SIAM uses two trust anchors: RPKI and the SCION CP-PKI. For property S1, the MS uses RPKI's ROAs. It queries RPKI repositories to ascertain that the *as* is authorized to originate IP prefix *ipPrefix*. This validation is performed when entities in its ISD submit $map_{ip,as}$ and when the MS pulls lists $list_{isd}$ from the PI. For property S2, the SCION CP-PKI is used to validate the origin AS of $map_{ip,as}$ is *as*. This ensures that ASes can only create mappings for themselves.

Downgrade Attacks. There are two entities that are in a position to perform downgrade attacks violating property S3 (i.e., convince SIGs that no mapping exists even though it does): PGNs and MSes.

A PGN could simply send empty tokens instead of an existing obj_{ms} in $\boxed{8}$. However, to prevent legitimate PGNs from receiving the correct list, an adversary would need to completely partition the gossip network, which is virtually impossible considering default values for k_{pgn} (see Table 2) and the fact that these must be located in different ISDs. Furthermore, a misbehaving PGN can be identified quickly based on the signatures on empty objects. A different possible attack is to provide an empty token to a querying MS in $\boxed{10}$. The MS defends against such attacks by requiring f_{ms} separate empty objects from PGNs in different ISDs. Furthermore, the MS can explicitly query PGNs in ISDs it trusts.

In SIAM, an MS can attempt to mount downgrade attacks *only* on ASes in its ISD by excluding a $map_{ip,as}$ either in $\boxed{4}$ or $\boxed{12}$. The first case—if an MS does not include a mapping submitted by a SIG in $\boxed{4}$—can be detected by the AS that originated $map_{ip,as}$. The integrity of individual mappings returned in $\boxed{12}$ can be verified by any SCION entity by checking CP-PKI signatures and querying the RPKI infrastructure using the same technique that MSes use to validate the mappings, see §5.3. If an MS excludes an existing mapping in $\boxed{12}$, this can be detected through out-of-bands communication and proven through the signatures on the messages.

DoS Attacks. Availability is a core requirement for SIAM (property S4) and it is protected through several mechanisms. Using a distributed PI enables an MS to query any of the PGNs in case some are under a DoS attack. Furthermore, to disrupt the gossip protocol, an attacker would have to partition the gossip network by attacking a large number of PGNs. The long validity periods of most SIAM entries (see Table 2) together with local storage at MSes and SIGs ensures that intermittent outages of individual SIAM components do not directly affect SIAM's operation. Amplification attacks exploiting the large size of PGN responses are prevented by verifying source CP-PKI signatures on requests. Finally, additional availability mechanisms proposed for SCION, including source authentication based on DRKey [24] and bandwidth guarantees with SIBRA [2], enables SIAM components to protect and authenticate traffic to ensure availability for legitimate requests, in particular as MSes only need to accept requests from SIGs within the same ISD.

Sybil Attacks. An adversary could attempt to create many additional PGNs to either partition the gossip network or provide the required number of empty objects to MSes. As both PGNs and

MSes require the PGNs they communicate with to be located in *different* ISDs and can prefer ISDs they trust, an attacker would need to either control core ASes in multiple existing ISDs or create new ISDs. As creating new ISDs requires a lot of effort in SCION, this would be infeasible in practice.

7 Evaluation

Scalability. We have chosen parameters k_{pln} and k_{pgn} equal to the Bitcoin network, which has been shown to be well connected for the number of nodes (∼10 000) that we consider as a maximum for SIAM (assuming 10 core ASes in each of 1000 ISDs) [8]. Our gossip networks (like the Bitcoin network) closely fit the model of "K-out graphs", which are virtually certain to be connected according to theoretical and empirical results [27]. Results from related random regular graphs suggest that, for these values, the network diameter is below 10 [4]. This means that new information reaches all gossip nodes within less than 2 h (even if the nodes are synchronized), which is in line with recommendations for the synchronization interval of RPKI of at most 4–6 h [5].

The system state and communication overhead increase linearly with the number of IP prefixes n_{p}. For each additional {IP prefix, AS} pair, there is an additional communication overhead of 231 additional bytes in gossip exchanges and mapping lists in our implementation. For $n_{\text{p}} = 10^6$ (the number of prefixes in the current Internet), the total state amounts to ∼230 MB, which is small enough to be transmitted in every gossip interval.

Implementation and Setup. We implement the different components of SIAM as SCION services. The services are written in Go and we use goroutines (lightweight threads managed by the Go runtime) wherever possible to allow for parallel execution. We use ECDSA signatures and SHA-512 for hashing; The services use SQLite 3 for local storage. Our prototype is integrated into the SCION open-source implementation and available online [28].

In our evaluation, we run services on the same machine when necessary and do not consider network traffic or bandwidth limitations. The experiments are run on a server with two Intel Xeon Gold 6242 2.8 GHz CPUs with 32 cores and 196 GB RAM in total.

Performance. We consider two main questions in our evaluation:

Q1 Does the gossip network scale to the size of the Internet?
Q2 Is the processing at MSes efficient enough in an Internet-scale deployment during partial deployment of SCION?

To answer Q1, we measure the time it takes for the gossip nodes to handle the different types of messages. The results of this measurement are shown in Table 3; they show that the gossip nodes take at most ∼3 seconds to process requests. This performance is acceptable since the gossip messages are propagated every 10 min and MSes pull lists from PGNs every hour (see Table 2).

An MS pulls mapping lists for all available ISDs every hour from the PI and should reverse the lists to form mappings from IP prefixes to AS numbers. To answer Q2, we evaluate the time it takes for the MS to reverse mapping lists (obj_{ms}) with a varying number of AS entries for (i) one ISD and (ii) 1000 ISDs in Table 4. We observe that the processing time scales linearly with the number of AS entries. The processing time of ∼5 min for the case of 1000 ISDs with 1000 entries each is acceptable considering the pull interval of 1 h.

Table 3: SIAM message-processing times including CP-PKI validation. Each ISD list has 1000 AS entries.

Messages	From	To	Time (ms)	Description
0a	PGN	PLN	17.6	1 PGN register request
0c	PLN	PLN	545	Gossip list of 1000 PGNs
2, 3	PGN, MS	PLN	139	Send list of 1000 PGNs
4, 5	MS	PGN	112	Process one ISD list
8	PGN	PGN	3250	Gossip 1000 ISD lists
9, 10	MS	PGN	1630	Send 1000 ISD lists

Table 4: Time to reverse mapping lists for 1 and 1000 ISDs including CP-PKI and RPKI validation.

No. of AS Entries/ISD	1 ISD (ms)	1000 ISDs (s)
10	35	8.05
100	80	49.3
1000	238	293
10 000	2630	2780

8 Related Work

Trotsky [20] and Plutarch [6] are architectural frameworks which intend to facilitate incremental deployments of new network architectures. Those frameworks embrace the fact that network architectures are heterogeneous and they improve interoperability between those different architectures. The SIG and SIAM do not strictly follow these frameworks but implement a similar idea for a concrete next-generation Internet architecture: SCION.

DANE [14] uses DNSSEC to bind server certificates that are used in TLS to domain names. Thus, it couples the DNSSEC PKI with the TLS PKI, similarly to SIAM coupling RPKI and SCION's CP-PKI. IPA [17] also leverages DNSSEC to provides a secure mapping between an AS public key and an IP prefix.

Passport [19] is a system that allows source addresses to be authenticated, preventing for example source address spoofing. This is achieved through creating pairwise shared keys between ASes through a key exchange piggy-backed on routing messages. Similar to SIAM using RPKI, it transfers security properties of the routing system to a new security protocol.

9 Conclusion

SIAM builds a bridge between the IP-based and the SCION Internet by making it possible to transfer authorizations from the RPKI to SCION. This allows the SIG—a transition mechanism that enables tunneling of IP packets through a SCION network—to be configured automatically according to publicly registered mappings of addresses. SIAM prevents hijacking attacks, is resilient to downgrade attacks, and avoids single points of failure. As such, SIAM meets the design goals of a responsible Internet—ensuring controllability, accountability, and transparency—and provides a backwards-compatible method of securing inter-domain communications.

For future work, we are considering various optimizations and extensions to SIAM including only transmitting hashes of lists in the gossip protocol and fetching full lists on demand; allowing the MS to renew lists that have not changed by pushing only signed hashes with new timestamps for freshness; and extending the PI to accept lists with a *type* tag so the PI can be used by different entities as a general distributed storage mechanism.

References

[1] Anapaya Systems. 2021. SCION-Internet: The New Way To Connect. https://www.anapaya.net/scion-the-new-way-to-connect.

[2] Cristina Basescu, Raphael M. Reischuk, Pawel Szalachowski, Adrian Perrig, Yao Zhang, Hsu-Chun Hsiao, Ayumu Kubota, and Jumpei Urakawa. 2016. SIBRA: Scalable Internet Bandwidth Reservation Architecture. In *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS)*.

[3] Tony Bates. 2021. CIDR Report. https://www.cidr-report.org/as2.0/.

[4] Béla Bollobás and W Fernandez De La Vega. 1982. The diameter of random regular graphs. *Combinatorica* 2, 2 (1982), 125–134.

[5] R. Bush. 2014. *Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI)*. RFC 7115. https://doi.org/10.17487/RFC7115

[6] Jon Crowcroft, Steven Hand, Richard Mortier, Timothy Roscoe, and Andrew Warfield. 2003. Plutarch: An Argument for Network Pluralism. *Computer Communication Review* 33 (01 2003), 258–266. https://doi.org/10.1145/972426.944763

[7] Alberto Dainotti, Claudio Squarcella, Emile Aben, Kimberly C Claffy, Marco Chiesa, Michele Russo, and Antonio Pescapé. 2011. Analysis of country-wide Internet outages caused by censorship. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*. 1–18.

[8] Varun Deshpande, Hakim Badis, and Laurent George. 2018. BTCmap: Mapping Bitcoin Peer-to-Peer Network Topology. In *IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*. 1–6. https://doi.org/10.23919/PEMWN.2018.8548904

[9] D. Dolev and A. Yao. 1983. On the security of public key protocols. *IEEE Transactions on Information Theory* 29, 2 (March 1983), 198–208.

[10] Dan Goodin. 2018. Google goes down after major BGP mishap routes traffic through China. https://arstechnica.com/information-technology/2018/11/major-bgp-mishap-takes-down-google-as-traffic-improperly-travels-to-china/.

[11] Google. 2021. Certificate Transparency: How CT works. https://certificate.transparency.dev/howctworks/.

[12] Gideon Greenspan. 2015. Avoiding the pointless blockchain project. https://www.multichain.com/blog/2015/11/avoiding-pointless-blockchain-project/, archived at https://perma.cc/GW53-2U9H.

[13] Cristian Hesselman, Paola Grosso, Ralph Holz, Fernando Kuipers, Janet Hui Xue, Mattijs Jonker, Joeri de Ruiter, Anna Sperotto, Roland van Rijswijk-Deij, Giovane C. M. Moura, Aiko Pras, and Cees de Laat. 2020. A Responsible Internet to Increase Trust in the Digital World. *Journal of Network and Systems Management* 28, 4 (2020), 882–922. https://doi.org/10.1007/s10922-020-09564-7

[14] P. Hoffman and J. Schlyter. 2012. *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*. RFC 6698. https://doi.org/10.17487/RFC6698

[15] Jonghoon Kwon, Juan A. García-Pardo, Markus Legner, François Wirz, Matthias Frei, David Hausheer, and Adrian Perrig. 2020. SCIONLab: A Next-Generation Internet Testbed. In *Proceedings of the IEEE Conference on Network Protocols (ICNP)*.

[16] M. Lepinski and S. Kent. 2012. *An Infrastructure to Support Secure Internet Routing*. RFC 6480. https://doi.org/10.17487/RFC6480

[17] Ang Li, Xin Liu, and Xiaowei Yang. 2011. Bootstrapping Accountability in the Internet We Have. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. https://www.usenix.org/conference/nsdi11/bootstrapping-accountability-internet-we-have

[18] Pat Litke and Joe Stewart. 2014. BGP Hijacking for Cryptocurrency Profit. https://www.secureworks.com/research/bgp-hijacking-for-cryptocurrency-profit.

[19] Xin Liu, Ang Li, Xiaowei Yang, and David Wetherall. 2008. Passport: Secure and Adoptable Source Authentication. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX Association, San Francisco, CA. https://www.usenix.org/conference/nsdi-08/passport-secure-and-adoptable-source-authentication

[20] James McCauley, Yotam Harchol, Aurojit Panda, Barath Raghavan, and Scott Shenker. 2019. Enabling a Permanent Revolution in Internet Architecture. In *Proceedings of the ACM Special Interest Group on Data Communication* (Beijing, China) *(SIGCOMM '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3341302.3342075

[21] George G. Michaelson, Geoff Huston, Tom Harrison, Tim Bruijnzeels, and Martin Hoffmann. 2021. *A profile for Resource Tagged Attestations (RTAs)*. Internet-Draft draft-ietf-sidrops-rpki-rta-00. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-rpki-rta-00 Work in Progress.

[22] Network Security Group, ETH Zurich. 2021. SCIONLab. https://www.scionlab.org/.

[23] Adrian Perrig, Pawel Szalachowski, Raphael M. Reischuk, and Laurent Chuat. 2017. *SCION: A Secure Internet Architecture*. Springer. https://doi.org/10.1007/978-3-319-67080-5

[24] Benjamin Rothenberger, Dominik Roos, Markus Legner, and Adrian Perrig. 2020. PISKES: Pragmatic Internet-Scale Key-Establishment System. In *Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIACCS)*. https://doi.org/10.1145/3320269.3384743

[25] Stephen Shirley. 2019. SCION ISD and AS numbering. https://github.com/scionproto/scion/wiki/ISD-and-AS-numbering.

[26] Aftab Siddiqui. 2021. A Major BGP Hijack by AS55410-Vodafone Idea Ltd. https://www.manrs.org/2021/04/a-major-bgp-hijack-by-as55410-vodafone-idea-ltd/.

[27] Mansi Sood and Osman Yagan. 2020. Tight Bounds for Connectivity of Random K-out Graphs. arXiv:2006.10638 [cs.IT]

[28] Supraja Sridhara. 2021. SCION open-source implementation including SIAM components. https://github.com/suprajasridhara/scion/tree/siam.

[29] Swisscom AG. 2021. Enhancing WAN connectivity and services for Swiss organisations with the next-generation internet. https://www.swisscom.ch/en/business/enterprise/downloads/security/international-connectivity-business.html.

[30] Andree Toonk. 2017. BGP leak causing Internet outages in Japan and beyond. https://web.archive.org/web/20170828092034/https://bgpmon.net/bgp-leak-causing-internet-outages-in-japan-and-beyond/.

[31] Brian Trammell, Jean-Pierre Smith, and Adrian Perrig. 2018. Adding path awareness to the Internet architecture. *IEEE Internet Computing* 22, 2 (2018), 96–102.