PrefiSec: A Distributed Alliance Framework for Collaborative BGP Monitoring and Prefix-based Security

Rahul Hiran
Linköping University
Sweden
rahul.hiran@liu.se

Niklas Carlsson
Linköping University
Sweden
niklas.carlsson@liu.se

Nahid Shahmehri
Linköping University
Sweden
nahid.shahmehri@liu.se

ABSTRACT

This paper presents the design and data-driven overhead analysis of PrefiSec, a distributed framework that helps collaborating organizations to effectively maintain and share network information in the fight against miscreants. PrefiSec is a novel distributed IP-prefix-based solution, which maintains information about the activities associated with IP prefixes (blocks of IP addresses) and autonomous systems (AS). Within PrefiSec, we design and evaluate simple and scalable mechanisms and policies that allow participating entities to effectively share network information, which helps to protect against prefix/subprefix attacks, interception attacks, and a wide range of edge-based attacks, such as spamming, scanning, and botnet activities. Timely reporting of such information helps participants improve their security, keep their security footprints clean, and incentivizes participation. Public wide-area BGP-announcements, traceroutes, and simulations are used to estimate the overhead, scalability, and alert rates. Our results show that PrefiSec helps improve system security, and can scale to large systems.

Categories and Subject Descriptors

C.2.0 [**Computer-communication Networks**]: General—*Security and Protection*; C.2.2 [**Computer-communication Networks**]: Network Protocols—*Routing protocols*

Keywords

BGP Monitoring; Prefix-based Security; Collaboration; Distributed Alliance Framework; Interception; Hijack

1. INTRODUCTION

Today, organizations and network owners must protect themselves against a wide range of Internet-based attacks. The Border Gateway Protocol (BGP) is susceptible to prefix hijacks, sub-prefix hijacks, and interception attacks [6, 7]. Edge networks and the machines within these networks may be scanned, probed, or spammed with unwanted traffic/mail [1,3,30]. In addition, network owners must be aware that machines within their networks may be compromised, participate in botnet activities, DDoS attacks, or in other ways cause harm.

Unfortunately, miscreants are becoming increasingly sophisticated and security attacks are no longer isolated events. Instead, attacks often cover multiple domains and behaviors, making them difficult to detect for a single network entity. Collaboration among network entities provides richer information, and can help detect and prevent such attacks [30, 31]. With an expected increase of cyber attacks and an urgent need for strengthened network security [34], it is important to design systems that help responsible organizations collaborate in the battle against miscreants.

While collaboration among organizations has been proposed, and the value of such collaboration demonstrated (e.g., [8, 22, 31]), it remains an open problem to design distributed mechanisms that provide effective decentralized information sharing among disparate organizations and Autonomous Systems (AS). In this paper, we present the design and data-driven overhead analysis of PrefiSec, a distributed system framework that (i) provides scalable and effective sharing of network information, (ii) provides notification alerts and aggregated evidence information about a wide range of attacks, and (iii) helps responsible organizations to keep their security footprints clean.

Scalable overlay design (Sections 2): At the center of our design is a distributed reporting and information monitoring system that allows participating members to effectively share route/prefix information and observations, report suspicious activities, and retrieve information about organizations, networks, their IP prefixes (blocks of IP addresses), and the activities within each prefix. To capture the intricate relationship structure between ASes and their prefixes, as well as the hierarchical nature of the IP space, we design an overlay consisting of complementary Distributed Hash Table (DHT) structures, and a novel distributed Chord [33] extension that provides functionalities such as longest-prefix matching, used in Internet routing.

Distributed alert mechanisms for prefix and subprefix hijacks (Section 3): BGP uses prefix announcements to determine the routing paths that will be taken by Internet Protocol (IP) packets. A (sub)prefix hijack involves an AS announcing a (sub)prefix allocated to another AS without permission. Building on our longest-prefix capable overlay, we design mechanisms for effective and distributed prefix- and subprefix-hijack attack detection and alert noti-

WISCS'14, November 3, 2014, Scottsdale, Arizona, USA.

http://dx.doi.org/10.1145/2663876.2663879.

fication. We provide the same notification accuracy of origin AS changes as existing central systems (e.g., PG-BGP [21] and PHAS [23]), but distribute the processing across all participants and avoid a single (trusted) point of failure, which typically see extremely high processing load [7].

Collaborative alert mechanisms for interception attacks (Section 4): Hijacked traffic is even more difficult to detect if the intercepted traffic is re-routed to the intended destination. As such interception attacks typically does not disrupt the service and involve many ASes, whose individual decisions can impact the success of the attacks [19], collaboration is important in detecting and defending against these attacks. Leveraging our overlay and the information that it maintains about AS relationships, we design simple policies and mechanisms for collaborative interception detection, which are low in overhead.

Aggregated prefix-based monitoring (Section 5): PrefiSec also provides effective mechanisms for monitoring and bookkeeping about a wide range of edge-network-based attacks, including scanning, spamming, DDoS attacks, and botnet activity. Our prefix-based structure effectively aggregates (often sparse) information from many reporters; e.g., about potential non-legit mail servers originating within a prefix. Such information can help responsible organizations keep their network security footprint clean. With malicious hosts increasingly alternating between malicious behaviors [22, 30], a combined per-prefix repository also helps improve early detection rates across services [31].

Data-driven overhead analysis: Throughout the paper we use public wide-area BGP-announcements, traceroutes, and simulations to estimate the overhead, scalability, and alert rates. Our analysis shows that our distributed solution is scalable, comes with low communication overhead, and allows participating organizations to improve their overall security. For example, our case-based study of the China Telecom incident (that occurred on April 8, 2010) shows that the system would have detected all hijacked prefixes, while maintaining relatively low per-node communication overhead and per-node processing and storage requirements; all non-increasing with increased alliance size. The paper is concluded with a review of related work (Sections 6).

2. SYSTEM OVERVIEW

The PrefiSec framework is an application layer service that leverages sharing of network activity observed by routers, network monitors, and other infrastructure. While our design allows both edge networks and ASes to join the alliance, for simplicity of presentation, we assume that a network is an AS with multiple prefixes. Like ASes, edge networks can have multiple prefixes. To map to our AS-focused presentation, edge networks are mapped under a single AS, making them responsible for a fraction of the AS's prefix space. Larger organizations that operate under multiple ASes can simply be considered as multiple members.

Figure 1 provides an overview of the PrefiSec architecture. Here, AS1-AS3 operate separate nodes in the PrefiSec overlay network. We assume that trusted personal relationships among network operators are used to create the overlay network. (A multi-tiered extension is also discussed.) PrefiSec is designed to effectively share and manage any information about ASes and their prefixes. As an example, we present mechanisms and policies designed to effectively detect and/or raise alerts about potential interdomain routing attacks. Relying primarily on reports about origin AS and AS-PATH announcements, we assume that each participating AS collects (e.g., [13, 14]) and share selected BGP updates from its edge routers, for example.

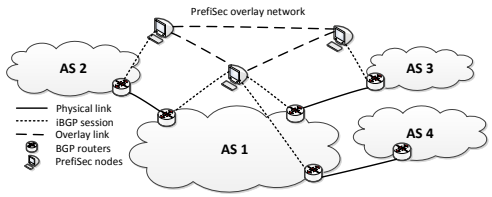


Figure 1: High-level PrefiSec architecture.

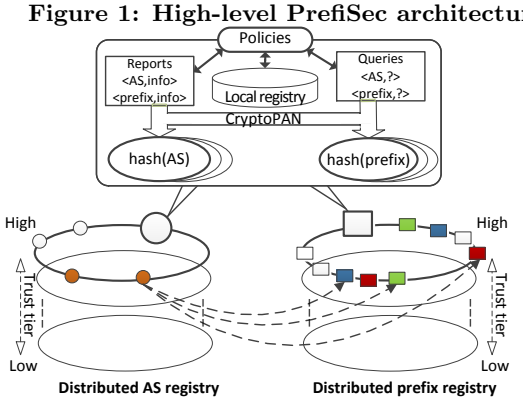


Figure 2: Overview of the framework, its key components, and structure.

2.1 Distributed overlay

Scalable overlay structures: To keep track of the activity associated with each organization and its IP prefixes, we maintain two complementary distributed structures.

- **Prefix registry:** We design a novel Chord-based [33] DHT, which stores prefix origin information (e.g., prefix-to-AS mappings) and observations of edge-network miscreant activities (e.g., scanning, spamming, etc.). The registry keeps track of the prefix hierarchy, and uses a distributed longest prefix matching algorithm for efficient insertion/retrieval.

- **AS registry:** A second Chord-based DHT is used to store information about ASes, their relationships, and AS-to-prefix mappings.

Figure 2 provides an overview of our PrefiSec framework, and shows how the two registry structures are linked by the prefix-to-AS and AS-to-prefix mappings (pointers in figure). Here, a participating member operates a node in the distributed AS registry and one node in the distributed prefix registry (e.g., the large circle and large rectangle, respectively, in the bottom-half of the figure) according to a set of built-in policies and locally stored/retrieved information (as shown in the upper-half of the figure). Reports and queries with shared information and observations are used to populate the registries. Incremental deployment is easily achieved by adding/removing nodes to/from these structures, as members join/leave the alliance.

Distributed information sharing and aggregation: Members share information about prefixes and ASes using *reports* directed to dedicated *holder* nodes (determined based on the reported AS or prefix). Each holder node is responsible for many ASes and prefixes, and for each AS or prefix,

the holders aggregate the information from many reporters. The holder nodes can help the other alliance members (i) by answering direct queries, and (ii) by creating and forwarding aggregated summary reports. Similar to publish-subscribe systems, members can also subscribe to summary reports. We expect that responsible organizations, wanting to keep their network security footprint as clean as possible, subscribe to their own prefix and AS information.

2.2 Distributed prefix registry

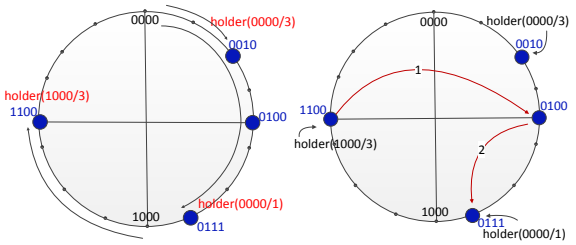
For our AS registry, we use Chord [33] more or less "out of the box". We pick a circular identifier space large enough to uniquely specify any AS (e.g., based on its AS number). For the prefix registry, on the other hand, Chord's (flat) circular identifier space does not naturally capture the hierarchical relationships between prefixes, and must be modified.

Ideally, prefixes of any length should be uniquely assigned to holders, and, given an IP address, the structure should return the holder of the longest-matching prefix. For example, for address 123.123.123.23, prefix 123.123.123.0/24 should be given priority over prefix 123.123.0.0/16. This section describes how we extend Chord to achieve unique and consistent longest-prefix-based assignment and lookup.

Longest-prefix discovery: Global IP-to-prefix lookup queries are resolved using a two-level greedy routing approach. At a high level, we first forward the query to the potential candidate holder h_k of the longest possible prefix of length k, if that prefix exists in the DHT. If h_k is not aware of such a prefix, it forwards the query to the next candidate holder h_{k-1}, which would be responsible for the next longest prefix (of length $k-1$), and so forth, until a prefix is found. For each such high-level forwarding step, multiple regular (low-level) Chord forwardings may be needed. Since /24 typically is the most specific prefix allowed by modern BGP routers, we use $k = 24$ as our initial choice for k.

Holder assignment: Our system defines the holder of a prefix as the node responsible for the last IP address in the prefix. Given a clockwise identifier space, only this choice ensures that the next candidate holder for a prefix of length $k-1$ is ahead of (or the same node as) the holder of the prefix of length k. With this selection of the holder node, in the majority of cases, the next candidate holder for a prefix of length $k-1$ is the same as the holder of the candidate prefix of length k (e.g., in 50% of the cases the last significant bit in the prefix of length k is a 1), and in the other cases, the next node is located in a region of the identity space for which the node has many shortcut pointers.

Example: Figure 3 presents a simple toy scenario, with a total identifier space of $2^4 = 16$ and four nodes: 0010, 0100, 0111, and 1100. Figure 3(a) shows how the prefixes 0000/3, 0000/1, and 1000/3 are assigned to the nodes 0010, 0111, and 1100, respectively. Figure 3(b) shows the high-level messages when node 1100 queries for the longest-prefix match for address 0011. In this case, node 1100 first uses Chord routing to route the query to the node (0100) responsible for the last address (0011) in the prefix 0011/4. When node 0100 receives this query, it observes that it does not have any entries for candidate prefixes 0011/4 and 0010/3, though it would be responsible for both. It then determines that the next biggest range is 0000/2 and uses Chord to route to the last address (0111) in this range. While node 0111 does not have an entry for 0000/2 it is in fact the holder of prefix 0000/1, and can resolve the original query.



(a) Holder assignment (b) Query forwarding

Figure 3: Holder assignment, prefix mapping, and longest-prefix query routing.

Reliability: To ensure efficient recovery at node departures, Chord typically copy the information stored at a node to its successor. For additional reliability, load balancing, and to ensure that no single node is responsible for the entire evaluation of a prefix, multiple holders per prefix are used. Figure 2 shows two holder nodes per AS (e.g., brown circles) and prefix (e.g., red rectangles). Here, CryptoPAN [12] is used to find additional holder nodes for each prefix.

CryptoPAN is a prefix preserving IP address anonymization scheme. With CryptoPAN IP addresses are mapped one-to-one in a manner such that two IP addresses that belong to the same /k-subnet also are part of the same /k-subnet in the new address space. This property allows us to ensure that the hierarchical features of our prefix registry are preserved when applying CryptoPAN to the original IP prefix (or address) in order to obtain H new keys (IP prefix). Using hash-based replication, load balancing is provided complementary. In general, nodes should query multiple holders and inform holders about potential inconsistencies, which may need to be resolved.

Local registry and optimizations: Two optimizations help reduce the Chord-related lookup overhead. First, each node maintains a local registry (Figure 2) with information about the prefixes and ASes that it sees, records statistics for these, and then informs the appropriate holder nodes. The system operates according to a soft-state protocol, with a time-to-live-based cache, and updates entries when changes are detected. A node that has out-of-date information can easily and quickly update its local registry (e.g., prefix tables) using the global DHT registries.

Second, when additional storage overhead is acceptable, existing 1-hop routing optimizations [17] can be used to reduce each lookup to a single hop. While such schemes require each node to have a pointer to every alliance member, Gupta et al. [17] show that the use of slice leaders allows timely, efficient, and scalable updating of the membership pointers and responsibilities under node churn, even for membership sizes up to a few million members. With much fewer existing ASes, and on the order of half a million routable prefixes, we foresee these optimizations to be feasible down to the granularity of ASes and the prefixes seen by most core routers. Appendix A presents a data-driven overhead analysis of our distributed prefix registry.

2.3 Policies and service implementation

Basic high-level services: Building on our scalable overlay, we present mechanisms and policies (Figure 2) that allow participating organizations to collaboratively detect and raise alerts about a wide range of attacks. Central routing-related detection mechanisms and policies are built

into the overlay itself, whereas high-level mechanisms and policies that help to provide additional services are built on top of the overlay, each leveraging the scalable system design. The system provides scalable detection and alert notification services for three broad classes of attacks: prefix and subprefix hijacks (Section 3), interception attacks (Section 4), and aggregated prefix-based monitoring (Section 5).

Service implementation: As part of providing the above high-level services, the prefix registry and AS registry also implement four effective distributed services that can be used as building blocks for these and other high-level services: (i) IP-to-prefix mapping, (ii) prefix-to-AS mapping, (iii) AS-to-prefix mapping, and (iv) other per-AS and per-prefix information extracted and stored in the repositories. The registries are updated as members observe new mappings, and the holder nodes can easily aggregate sparse information; e.g., to identify and store information about ASes that likely are Internet eXchange Points (IXPs) or siblings. In the following, we explain how both high-level services and these basic building blocks are designed.

Incentive-based hierarchy extension: While our design easily extends to a multi-tiered trust hierarchy (in which nodes are promote/demoted between tiers based on their reporting [11], for example), in the following we assume that all nodes belong to the same tier (setup based on trusted personal relationships, for example), and focus on the scalability and overhead of the system design.

3. PREFIX AND SUBPREFIX HIJACKS

In contrast to the central processing of prefix origin history used by systems such as PG-BGP [21] and PHAS [23], our system distributes the responsibility and processing of prefixes among holder nodes. These nodes act as information aggregators that maintain history for each prefix, allowing us to improve the scale and accuracy compared to what is possible with central approaches. By distributing the responsibility across multiple holders, PrefiSec also avoids a single point of failure or trust.

3.1 Policy overview

Prefix hijack: We design a distributed prefix hijack detection policy based on PHAS [23]. As discussed in Section 2, each participating organization operates one *node* in the AS registry and one node in the prefix registry. The *holder* node of each prefix performs information aggregation and evaluation for that prefix, but is also responsible for detecting when there are changes in the origin AS for a prefix, as well as notifying the previous origin AS of the prefix when a new AS claims ownership of the prefix.

An overview of our hijack alert notification policy is given in Figure 4. The policy is invoked at a node in the alliance network when it sees a new prefix p, a new origin for a prefix p, or when the TTL for the prefix p expires (step 1). The node prepares a query with this information for the holder of prefix p (step 2), and the query is forwarded to the holder of prefix p (step 3) over the overlay network (Section 2.2).

For each prefix p, the holder node tracks the ownership set $A_p(t)$ over some time window of duration T. If the holder sees a change in the origin set, the current owner(s) of the prefix are notified and the ownership set $A_p(t+\epsilon)$ updated (step 4). The case when the prefix has not been previously observed is treated as a case of a potential subprefix hijack and the subprefix hijack policy is invoked at such times.

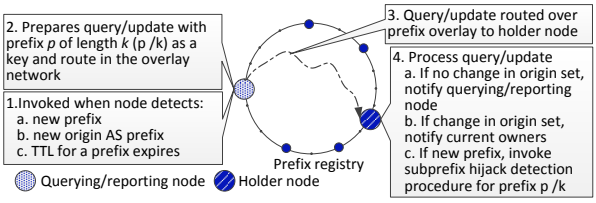


Figure 4: Prefix hijack alert notification policy

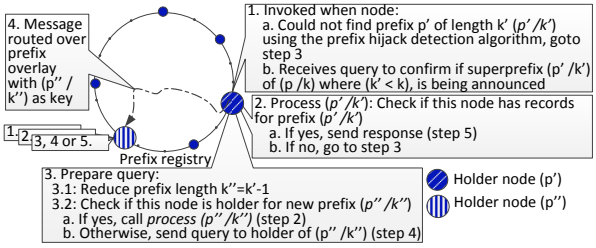


Figure 5: Subprefix hijack alert notification policy.

Subprefix hijack: When a prefix is observed for the first time, it is important to determine what less specific prefix this may be subprefix hijack attack on. We refer to such a prefix as a superprefix of the newly observed prefix. At the time of such occurrence, our distributed policy finds the immediate superprefix of the announced subprefix and notify the origin AS for the superprefix about the announcement. The origin AS for the superprefix is typically in the best position to determine if the announcement is part of a subprefix hijack attack, or whether the announcement is legitimate and authorized by the origin AS of the superprefix.

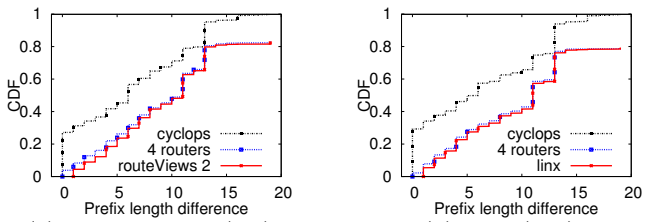
Figure 5 provides an overview of our subprefix hijack alert notification policy. The subprefix hijack policy is invoked by a holder node $h_{p'}$ when it receives a prefix query for prefix p' and does not have an entry for this prefix (step 1a). At this time, holder node $h_{p'}$ creates a superprefix query (step 3) and uses Chord (step 4) to send it to the next potential candidate, if needed. To find the next node to forward the query (step 3), the holder $h_{p'}$ reduces the prefix length, say k', of prefix p' by 1. Say prefix p'' is the new prefix with prefix length k'' (step 3.1), the holder node then checks if it is the holder for the new prefix created (step 3.2).

When the query arrives at this holder node, it again invokes the subprefix hijack detection procedure (step 1b). The new holder node checks if it has records for the new prefix (step 2). If the queried holder node has a record for the new prefix, the holder node will send the response and quit the procedure (step 2a). However, if it is not the holder, a new query will be prepared (step 3) that will be routed over the overlay network, with the new prefix p'' as the key (step 4). The process continues recursively until the superprefix p'' for prefix p' is found. When such superprefix is found, the holder node $h_{p''}$ reports owner set $A_{p''}(t-\epsilon)$ for prefix p'' about subprefix p' and the claimed origin set $A_{p'}(t+\epsilon)$.

3.2 Case-based overhead analysis

For our analysis, we examine the announcements seen around the time of the China Telecom incident [19] (April 8, 2010). This day, China Telecom announced origin of approximately 50,000 prefixes originated by others.

Giving consideration to the overhead both when networks are under attack and under normal circumstances, we use the routing tables and updates seen at all six servers participating in the Routeviews project during the first two weeks



(a) RouteViews 2 (US) (b) Linx (UK)

Figure 6: Cumulative Distribution Function (CDF) of the distance to the closest prefix in the global prefix registry, of newly observed prefixes.

of April, 2010. We base our original ownership lists on the RIB table data from April 1, and then use the BGP updates observed during the following period.

Table 1 shows results for April 7 and 8, the day before and the day of the incident, respectively. Results are presented for increasingly large alliances, each obtained by adding servers to the alliance, in the following order: RouteViews 2, Linx, Paix, Dixie, RouteViews 4, and Equinix.

Normal conditions: The traffic overhead is very small compared to that of PHAS and other techniques that would use central processing. For example, on April 7, 2010, PHAS would have required all 23 million announcements to be forwarded to and processed on a single node (totaling 867MB compressed or 3GB uncompressed data, if using all six monitors). The load scales proportionally with more members. In contrast, with PrefiSec, a particular node, say RouteViews 2, would make 209 prefix queries (due to prefixes with a new origin: "origin not seen") and 1,949 subprefix queries (due to new prefixes seen: "prefix not seen") to the overlay.

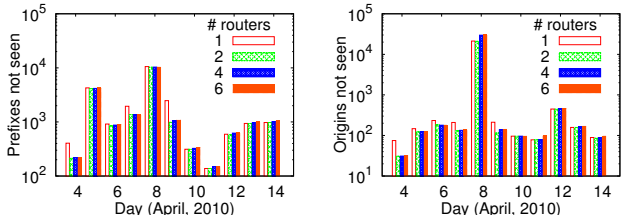
Furthermore, with six alliance members, out of all queries generated by individual nodes, 138 and 1,354 queries would eventually result in prefix and subprefix alerts, respectively. These results show that the number of alerts processed by holders scales very nicely with the alliance size. In fact, with the corresponding sub-prefix policy invocations being distributed across holder nodes, we note that the alerts generated per holder node reduce even more (faster than $1/N$). This additional reduction is achieved by distributed information aggregation at holders.

Figure 6 characterizes the overhead of such prefix insertions, as measured by the distance (in prefix lengths) between the two holders for prefix p (to be inserted) and the longest-matching prefix p' for which prefix p is a subprefix. We use three reference baselines: the RIB of the server itself, the combined RIBs of four different servers, and the global Cyclops database. We note that prefix length differences can be substantial, but decrease with larger alliance sizes.

Referring to Table 1, we can also see that the number of updates to the registry if using a (small) 24-hour window is much greater than if also taking into account the RIB information one week earlier (as per the much smaller values for the "origin not seen" statistics). Of course, using an adaptive window approach may lead to additional improvements [23].

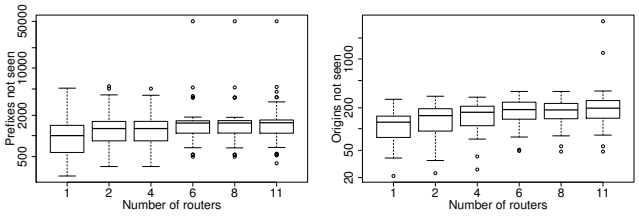
Day of incident: Our overlay allows effective collaborative detection of prefix and subprefix attacks. In fact, during the day of the incident the alliance would raise 40,675 alarms, including alarms for all 39,094 unique prefixes that had the specific signature associated with the incident [19].

Figure 7 shows that there is a significant increase in traffic overhead on the day of the incident (April 8, 2010), but that



(a) Prefix not seen (b) Origin not seen

Figure 7: Time-line of anomalous origin reporting.



(a) Prefix not seen (b) Origin not seen

Figure 8: Impact of alliance size.

the reporting overhead quickly decreases after the incident. We also note that our system would easily handle such an increase. First, only the prefix holders would need to communicate with the owners of the hijacked prefixes. Second, the holders can easily and quickly sanity check the claims, using the AS registry. China Telecom would have quickly been flagged and additional care could be taken until authenticity had been confirmed or a certain period of time had elapsed. Finally, as seen by the smaller "Prefix not seen (unique ASes)" statistics, the number of alerts to be sent can be reduced by aggregating messages to the same AS. For these statistics, the superprefix was found using Cyclops data and mapped to an AS using the RIPE whois database.

Present day (April 2014): Figure 8 presents per-day statistics from the eleven (11) monitors that remained active throughout April 2014, including the subset used for the 2010 data. While we observe large day-to-day variations during the month (logarithmic y-axis), it is encouraging that the average node generate on average less than 2K queries per day (not shown) and the number of subprefix hijack alerts (Figure 8(a)) and prefix hijack alerts (Figure 8(b)) scales very nicely with the number of members. For example, the average number of alerts changes from 1,153 to 3,246, and from 125 to 327 for the two types, respectively, as the number of members increase from one to eleven.

Keeping in mind that the storage overhead and number of prefixes (Section A) that each holder node is responsible for decreases in inverse proportion to the number of alliance members, we note that the queries processed per alliance node remains roughly constant, as this directly cancels the linear increase in the number of original queries generated by the entire alliance. In fact, with sub-linear increase in the number of alerts, it can be argued that the overhead per node decreases with growing alliance sizes.

4. INTERCEPTION ATTACK

One of the harder problems with BGP security is the detection of interception attacks [6, 7]. Figure 9 shows an example. Here, AS B announces that it is one hop away from C, although in reality, it is not connected to C. This announcement will not result in any prefix origin triggers, but may still allow B to intercept traffic on its way to C.

Table 1: Summary of route announcements statistics for a April 7 and April 8, 2010.

Metrics	April 7, 2010 (day before incident)				April 8, 2010 (day of incident)			
	1 node	2 nodes	4 nodes	6 nodes	1 node	2 nodes	4 nodes	6 nodes
Route announcements	9,179,307	16,000,217	18,912,643	23,206,562	10,177,068	19,371,838	23,388,253	31,265,557
Announced prefixes	323,899	327,400	328,630	328,826	331,348	332,922	333,825	336,526
Prefix not seen	1,949	1,367	1,349	1,354	10,554	10,346	10,332	10,330
Prefix not seen (unique AS)	290	241	242	247	273	250	261	263
Origin not seen	209	130	134	138	21,275	21,001	29,704	30,245
Origin not seen in last 72h	1,302	3,594	3,874	2,735	21,570	21,970	31,108	31,680
Origin not seen in last 24h	3,200	5,840	6,184	4,746	22,561	23,027	32,382	33,937

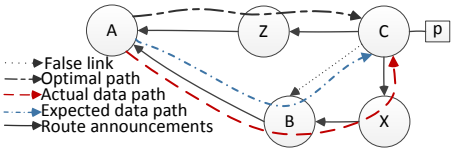


Figure 9: Detecting route inconsistencies.

As of today there is no straightforward way to automatically detect interception attacks. Instead, network owners must typically manually analyze and resolve suspicious inconsistencies between announced BGP AS-PATHs and the actual data paths. This section describes how PrefiSec can be used to reduce the number of suspicious inconsistencies.

4.1 Policy overview

We envision that members will maintain a history of the announced AS-PATHs, and evaluate any newly observed path-prefix pairs for inconsistencies. At such times, the member node (1) performs a traceroute to an IP address within the prefix, (2) uses the prefix registry to create a traceroute AS path [28], and (3) compares the announced AS-PATH (control-plane information) with the traceroute AS path (data-plane information). If the traceroute AS path does not match the announced AS-PATH, (4) the node uses information maintained by the AS registry regarding legit path discrepancy reasons. Finally, if no legit reason is found, (5) the node raises an alert and informs the appropriate AS and prefix holder nodes.

4.2 Legit path discrepancy reasons

To reduce the number of false alerts, it is important to keep track of legit reasons for suspicious path discrepancies between the announced AS-PATHs and the actual data paths. Figure 10 summarizes some common legit reasons for such differences [28]. We next describe how the AS registry can maintain information about such reasons.

IXP cases (Figure 10(a)): Internet eXchange Points (IXPs) [28] may cause extra hops in the traceroute path, not seen in the announced AS-PATHs. Extending the approach by Mao et al. [28], nodes that detect an extra AS hop X can report the ASes before and after X to the holder of X. This node can then calculate the number of unique ASes appearing just before and after X, referred to as the fan-in and fan-out factor, respectively. If these factors are greater than some threshold, the holder can classify X as an IXP.

MOAS cases (Figure 10(b)): In certain cases we may observe that AS X in the AS-PATH is replaced by AS Y in the traceroute path. Such replacement is common when the prefix is originated by multiple ASes (MOAS). We note that such MOAS cases are an artifact of mapping from an IP address to AS and not a result of a routing anomaly and can

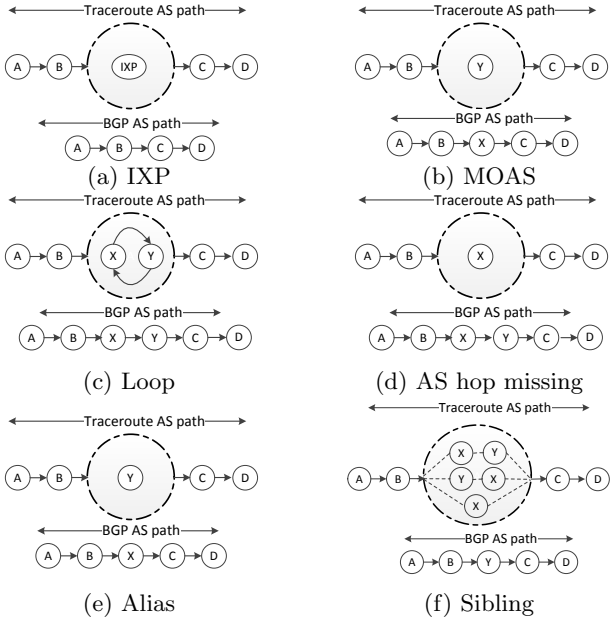


Figure 10: Legit reasons for path discrepancies.

be identified by holder nodes. Holders in the prefix registry can be informed about multiple co-origins, as described for IXPs, which could inform the AS holders about these relationships. Alternatively, the AS holders themselves can keep track of replacements reported by member nodes.

Loop cases (Figure 10(c)): Some traceroute paths exit and enter an AS more than once [28]. For example, an announced AS-PATH {A,B,C,D} may have a corresponding traceroute path {A,B,C,B,C,B,C,D}. These cases do not require any additional information from the AS registry.

Missing-hop cases (Figure 10(d)): Occasionally, an AS hop seen in the AS-PATH is not observed in the traceroute path. For example, in Figure 10(d), AS Y is missing in the traceroute path. This can occur for reasons such as routers in Y not responding to traceroute queries or using IP addresses from their neighbors. This case typically does not require any additional AS registry information, although it would be easy to add more AS information to the holder.

Alias cases (Figure 10(e)): When an AS X in the AS-PATH is replaced by an AS Y in the traceroute path, it may be due to a router having IP addresses from two different ASes on its interfaces. Such an IP address, called an alias address, may arise due to third-party address issues [29]. The alliance nodes can use existing third-party address detection methods [29], and report its findings to the holder node of the replacement AS hop Y in the AS registry, which can apply a threshold-based policy on the number of occurrences required for X and Y to be classified as an alias pairing.

Table 2: Reduction in the number of traceroutes.

Extra history (hours)	Server		
	Telstra	Global	Hurricane
2h	25.8%	25.1%	24.3%
4h	31.4%	32.6%	33.5%
8h	34.9%	38.2%	38.9%
16h	47.4%	45.6%	47.1%
24h	49.9%	48.6%	50.4%
48h	53.6%	51.9%	53.4%

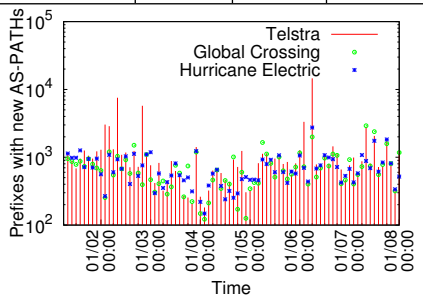


Figure 11: New routes observed (Nov. 1-7, 2012).

Sibling cases (Figure 10(f)): Other potential causes for valid discrepancies are route aggregation and sibling ASes, owned by the same organization [28]. Figure 10(f) illustrates a case in which the AS-PATH is $\{A,B,Y,C,D\}$ and ASes X and Y are sibling ASes. In the traceroute path we may observe Y being replaced by any of the following: $\{X,Y\}$, $\{Y,X\}$, or $\{X\}$. When an alliance node encounters such a case, it will report the AS-hop before and after the two-hop segment $\{X, Y\}$ in the traceroute path to the holder nodes of both X and Y. Similar to in the IXP case, if the fan-in and fan-out exceeds a threshold, the holder node detects a sibling relationship [28], and can inform the other holder.

4.3 Case-based analysis

We next consider how an AS can use the information provided by PrefiSec to identify suspicious and non-suspicious path inconsistencies. For this evaluation, we use measurements from three public Routeviews monitors and three nearby public traceroute servers, each pair hosted by Global Crossing (AS 3549), Telstra (AS 1221), and Hurricane Electric (AS 6939). These servers are located in Palo Alto (CA), Sydney (Australia), and San Jose/Livermore (CA).

Traceroutes: As with the prefix hijack detection overhead, great reductions in the number of traceroutes that must be executed can be achieved using a short history of previously seen AS-PATHs. Table 2 shows the reduction in the number of traceroutes to execute, when considering multiple RIB snapshots (each two hours apart), at each of the three servers. We observe diminishing returns, with most of the advantages obtained using a relatively small history. In the following we use a 24-hour history (49% reduction).

Figure 11 shows the number of traceroutes that would have to be executed, as a function of time, during the first week of November 2012. The three servers observe similar variations in the number of traceroutes they need to perform at each instance during this period, although the pairwise Pearson correlation factors (TG: 0.403; TH: 0.604; HG: 0.661) suggest that the correlation is only moderate.

Path comparison: For our analysis, we first convert the IP-level traceroutes to AS-level traceroutes. While our prefix registry is designed to provide this mapping, for the purpose of our evaluation, we used the Cymru whois database.

Table 3: Path comparison analysis (Nov. 1-7, 2012).

	Telstra	Global	Hurricane
Announcements	$3.6 \cdot 10^7$	$3.5 \cdot 10^7$	$3.6 \cdot 10^7$
Traceroutes (new route)	102,689	63,434	60,628
No data (new route)	506	60,045	3,200
Successful traceroutes	102,183	3,389	60,627
Direct matches	12,387	704	16,374
Subset matches	62,952	947	13,267
IXP matches	2,672	11	30,071
MOAS matches	309	NA	445
Loop matches	2,271	108	1,021
Missing hop matches	2,730	689	6,886
Alias matches	11,650	276	7,020
Sibling matches	1,764	27	243
Past matches	4,209	363	1,916
Future matches	487	23	515
Unresolved traceroutes	3,333	244	9,464
Unresolved triples	539	82	1,422

Nodes not responding to ICMP queries are considered as wild-cards ($*$) between the neighboring ASes.

As mentioned, there are many valid reasons why a traceroute path will not be a *direct match* to the announced AS-PATH. First, the IP-to-AS mapping could be incorrect or out-of-date. While our prefix registry will help, there is no 100% up-to-date information source for this mapping. Second, it may not be possible to map all routers along the paths, as some routers (wild-card nodes) may have disabled ICMP, or traceroute servers (in this case out of our administration) may terminate at some timeout value. We refer to queries that matched all observed ASes as *subset matches*. This scenario was common for the Telstra servers.

Third, our AS registry keeps track of AS relationship information about the six legit reasons for route anomalies described in Section 4.2. We call paths that match after applying each such condition (sequentially): *IXP matches*, *MOAS matches*, *loop matches*, *missing hop matches*, *alias matches*, and *sibling matches*. To approximate the conditions that a large-scale system would see, we populate the AS registry using public data, including known IXP prefixes [4], aliases from the iPlane project [27], and sibling relationship data from CAIDA [26]. MOAS cases are identified using IP-to-AS mappings from the whois Cymru database.

Finally, the announced AS-PATHs and traceroute paths may change over time, and may not always be in sync. Using a 2-hour window of path announcements seen at the server, we identify *past matches* and *future matches*. Past matches captures fluctuations in AS-PATHs. The future matches results from cases in which the announced path changes have not yet propagated all the way to the monitor server. If employed, the future match policy would of course require a time delay (e.g., two hours) before classifying a path.

Table 3 provides a breakdown of the traceroutes that we performed during the first week of Nov. 2012. We applied each rule in order, such that only traceroutes that did not match the previous criteria were considered for each new row. The *no data* cases correspond to cases in which the public traceroute server did not respond to our traceroute queries. The high number of such queries to the Global Crossing server is due to server limitations, but is not expected to have introduced biases affecting our results.

We note that the number of traceroutes that we could not automatically confirm (3,333; 244; and 9,464) is much smaller than the original number of successful traceroute queries (102,183; 3,389; and 60,627). We also see that IXP,

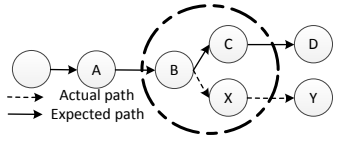


Figure 12: ASes involved in the anomaly

Table 4: Redundancy in unique triples.

				Redundancy in triples		
	Server	Traceroute	Triples	History	Sharing	Hybrid
Week 1	Telstra	66,985	372	–	12%	–
	Global	55,144	292	–	15%	–
	Hurricane	–	–	–	–	–
Week 2	Telstra	102,689	539	32%	14%	41%
	Global	63,434	82	27%	26%	38%
	Hurricane	60,628	1,422	4%	6%	7%
Week 3	Telstra	77,012	492	45%	16%	54%
	Global	–	–	–	–	–
	Hurricane	56,518	1,244	35%	6%	39%

alias, MOAS, and sibling matches are each responsible for a significant portion of the reduction of unique candidate announcements, validating the importance of the AS registry.

Further reductions: To further reduce the candidate cases to consider, we identify unique triples [6]. Referring to Figure 12, such a triple consists of (i) AS B that is responsible for the anomaly, (ii) AS C that B claims it will use, and (iii) AS X over which it actually forwards the packet. The last line in Table 3 shows that grouping into triples allows a 65-85% reduction in the number of cases to consider, as the reasons for these anomalies are often the same.

For the remaining candidate anomalies, nodes can contact the holders of AS B in the triple. If the holder node has seen this anomaly before and has found no problem, then it can respond that the anomaly is indeed benign. Otherwise, the holder node may need to do additional analysis.

The holder nodes are in a great position to take advantage of aggregate information. Table 4 shows the summary statistics for three basic aggregation and history-based approaches. The history column shows the percentage of triples that have also been observed by the same server in past weeks. The sharing column shows the percentage of triples that are also seen by at least one of the other servers (one or two, depending on the week). Finally, the hybrid column shows the percentage of triples that apply to at least one of the rules. We see that with just three members the number of triples that must be processed can be reduced by up to 43% (week 3). Clearly, there are significant advantages to maintaining history at the holder nodes, and sharing information across organizations.

We note that none of our policies or mechanisms thus far has relied on any knowledge of AS-to-AS relationships. An AS also wanting to investigate the *cause* for the remaining unresolved anomalies, may want to make use of AS-to-AS relationships such as customer-to-provider (C2P), peer-to-peer (P2P), and provider-to-customer (P2C) [19]. Unfortunately, most AS does not want to share their peering policies. To incorporate a classification of AS-to-AS relationships into the AS registry, future work could therefore include the design and evaluation of a distributed version of the valley-free [28] classification method by Shavit et al. [32].

Overhead discussion: Referring back to our interception detection and alert notification policy (Section 4.1), we note that all traceroutes (step 1) and path comparisons (steps 3) are done locally at the detecting node. Holder nodes are responsible for raising alerts and working with the identified triples (step 5). The overlay is only invoked when doing IP-to-AS mappings (step 2) and using AS relationships to reduce the number of mismatches (step 4). Of these, step 2 involves the most communication. For example, during the first week of Nov. 2012, our policy performed 14,670 traceroutes per day at the Telstra server and only 3,907 of these needed to be considered for further reductions after removing direct and subset matches. In contrast, 265,805 IP addresses where observed per day in the traceroutes.

Fortunately, out of the observed IP addresses, we observed only 8,675 unique IP addresses per day on average. Furthermore, if considering unique /24 prefixes, we could further reduce the number of queries to 3,967, and finally if also allowing a 24 hour rolling window of observed /24 prefixes this number is reduced to 2,787. The overlay should easily handle these lookups, and the processing load on each holder node does not change with the number of members (as the increased total load is distributed over more holders). The only increase in load that results from increasing membership is due to longer Chord routes. However, as discussed in Section 2.2, these overheads can easily be minimized based on if storage or forwarding costs are the primary bottleneck.

5. PREFIX MONITORING

In addition to routing-aware information, our prefix registry is designed to help organizations share anomaly alerts and effectively aggregate (often sparse) information about a wide range of attacks associated with edge networks and their prefixes. Shining the light on the organizations and network owners themselves, has many advantages. For example, using timely reports from other alliance members, the network owners can police miscreant activity within their own networks, clean up their network footprint, and ensure that compromised machines do not cause prolonged harm [8]. The use of prefixes also allows the system to scale effectively, and helps capture effects of subnetwork-aware IP-spoofing [3] and address migration due to dynamic address allocation. We use four basic examples to illustrate the power and generality of the framework.

Scanning attacks: To detect scanning, organizations typically monitor the incoming (and outgoing) traffic using intrusion detection systems (IDS) and classify each connection as either good or bad. Allman et al. [1] propose a set of heuristics to classify hosts based on the mix of good/bad connections. Our system is well suited to implement generalizations of such classification policies [3], in which prefixes are evaluated rather than hosts.

Spam server activity: Similarly, organizations can monitor SMTP traffic from non-legit servers and report such activity to the holder nodes. This information can then be used to inform the origin owner of the prefix of suspicious activity, and/or provide organizations with information about the general trust level associated with different prefixes. We note that this approach naturally extends to many other activities, including the presence of botnet servers.

Cross-class detection: It is common for malicious hosts to alternate between different malicious behaviors [31]. The holders have access to information related to many different types of attacks associated with a prefix, and is therefore in a good position to perform cross-class detection.

Attack correlation: Finally, networks are typically not attacked at random and often attacks are not isolated [30].

Targeted networks might therefore often benefit from additional information sharing and collaboration. Holders can act as matchmakers, informing victim networks about correlated networks that see similar attacks.

6. RELATED WORK

DHTs and DHT-based applications: Distributed Hash Tables (DHTs) such as Chord [33], have been used to improve the scalability for a wide range of distributed systems, including for co-operative web caching [20] and publish-subscribe systems [5]. To the best of our knowledge, this is the first work to allow the use of IP-prefix ranges.

Collaboration: Collaboration has been shown to help protect against different types of edge-network attacks [38], including DDoS attacks, system intrusions, and scanning. With these approaches, IDS monitors typically share data plane information and malicious IP addresses [38]. Collaborative fault detection has also been proposed for BGP [18]. With NetReview [18] BGP routing messages are recorded in tamper-evident logs that can be shared with others. Our systems are complementary, as NetReview could be used to share richer information to carefully analyze suspicious activity identified using our system, for example.

Central solutions: Both BGPMon[1] and Team Cymru[2] collect routing information from distributed monitors, and create alerts/summary reports about routing anomalies to which organizations can subscribe. Although Team Cymru provides a DNS-based lookup service for origin ASes, prefixes, and allocation dates, the service and all processing is centralized under a single administrative domain. In this paper we propose and evaluate a distributed solution.

Crypto-based architectures: The Resource Public Key Infrastructure (RPKI) [25] builds a formally verifiable database of IP addresses and AS numbers, and can be used to verify that the AS originating a prefix is authorized. While some recent works (e.g., [10]) have identified significant issues with the hierarchical RPKI management and the control it can give some entities (e.g., RIRs) on the global Internet routing (which can have significant political and business implications), RPKI also has many nice features, including incremental deployment (after some router software updates) and the ability to block hijacks. Proposals such as BGPsec [24] extend RPKI, to protect the AS path attribute of BGP update messages. Rather than blocking specific attacks, PrefiSec provides a scalable, fully distributed architecture for sharing of information that can be used to protect against many types of attacks.

Reverse DNS: ROVER [16] is a complementary approach to cryptographically secure BGP route origin, which builds on DNSSEC [2] and reverse DNS services. Despite years in development, less than 0.5% of .com and .net domains had signed up for DNSSEC by May 2014.[3] Requiring manual configuration, the future adoption rate of ROVER is unclear. In addition, recent revelations about government sponsored online surveillance have raised concerns regarding systems that require centralized root key distribution.

Hijack detection: There has been no large-scale deployment of crypto-based solutions [7]. Instead, both data-plane based [36, 37] and control-plane based [21, 23] techniques have been proposed for anomaly detection in BGP routing. As explained in Section 3, we extend existing control-plane based protocols (PG-BGP [21], PHAS [23]) for prefix ownership. Similar to Ballani et al. [6], we combine data-plane and control-plane information to detect and classify routing anomalies, but, in contrast, consider the more general case in which the anomaly can occur in the middle of the path. The combination of control-plane and data-plane data have been used to evaluate the accuracy of AS-level topology inference [35] and detecting hidden areas of the Internet [9]. These works have provided additional insights to our design.

[1]BGPMon, `http://www.bgpmon.net/`, May 2014.

[2]Team Cymru, `http://www.team-cymru.org`, May 2014.

[3]Verisign labs scoreboard, `http://scoreboard.verisignlabs.com/`, May 2014.

7. CONCLUSIONS

This work presents the design and data-driven overhead analysis of PrefiSec, a distributed system that provides scalable and effective sharing of network information, for the purpose of helping organizations detect and protect against prefix/subprefix attacks, interception attacks, and a wide range of edge-based attacks. We present a novel distributed solution, which maintains information about the activity associated with blocks of IP addresses (prefixes) and autonomous systems (AS). Our solution extends Chord [33], leverages unique properties of CryptoPAN [12], and implements new scalable mechanisms and policies for efficient information sharing. Using public wide-area BGP-announcements, traceroutes, and simulations, we show that the system is scalable with limited overhead. The system helps participants improve their security and keep their own security footprints clean. Our distributed mechanisms infer AS relationships from publically available information, including public route announcements, and participating ASes are not expected to share any information about their own networks and AS relationships. Of course, further improvements can be achieved if ASes also share some private information. Future work include the design and evaluation of sharing policies and incentive mechanisms, leveraging our incentive-based hierarchy extension.

8. REFERENCES

[1] M. Allman, V. Paxson, and J. Terrell. A brief history of scanning. In *Proc. ACM IMC*, 2007.

[2] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035 (Proposed Standard), 2005.

[3] M. Arlitt, N. Carlsson, P. Gill, A. Mahanti, and C. Williamson. Characterizing intelligence gathering and control on an edge network. *ACM TOIT*, pages 2:1–2:26, 2011.

[4] B. Augustin, B. Krishnamurthy, and W. Willinger. IXPs: mapped? In *Proc. ACM IMC*, 2009.

[5] R. Baldoni, C. Marchetti, A. Virgillito, and R. Vitenberg. Content-based publish-subscribe over structured overlay networks. In *Proc. IEEE ICDCS*, 2005.

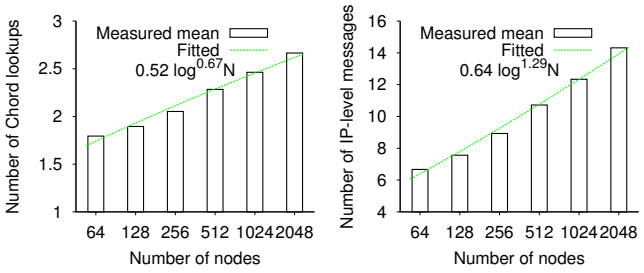
[6] H. Ballani, P. Francis, and X. Zhang. A study of prefix hijacking and interception in the Internet. *ACM CCR*, pages 265–276, 2007.

[7] K. Butler, T. Farley, P. McDaniel, and J. Rexford. A survey of BGP security issues and solutions. *Proceedings of IEEE*, 98:100–121, 2010.

[8] N. Carlsson and M. Arlitt. Leveraging organizational etiquette to improve Internet security. In *Proc. IEEE ICCCN*, 2010.

[9] K. Chen, D. R. Choffnes, R. Potharaju, Y. Chen, F. E. Bustamante, D. Pei, and Y. Zhao. Where the

sidewalk ends: Extending the Internet AS graph using traceroutes from p2p users. In *Proc. CoNEXT*, 2009.
[10] D. Cooper, E. Heilman, K. Brogle, L. Reyzin, and S. Goldberg. On the risk of misbehaving RPKI authorities. In *Proc. ACM HotNets*, 2013.
[11] C. Duma, N. Shahmehri, and G. Caronni. Dynamic trust metrics for peer-to-peer systems. In *Proc. PDMST*, 2005.
[12] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon. Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme. *Computer Networks*, pages 253–272, 2004.
[13] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. In *Proc. ACM SIGCOMM FDNA*, 2004.
[14] R. Fernando and S. Stuart. BGP Monitoring Protocol. IETF, 2012.
[15] P. Garca, C. Pairot, R. Mondjar, J. Pujol, H. Tejedor, and R. Rallo. PlanetSim: A new overlay network simulation framework. In *Proc. SEM*, 2004.
[16] J. Gersch and D. Massey. ROVER: Route origin verification using DNS. In *Proc. IEEE ICCCN*, 2013.
[17] A. Gupta, B. Liskov, and R. Rodrigues. Efficient routing for peer-to-peer overlays. In *Proc. NSDI*, San Francisco, CA, 2004.
[18] A. Haeberlen, I. Avramopoulos, J. Rexford, and P. Druschel. NetReview: Detecting when interdomain routing goes wrong. In *Proc. NSDI*, 2009.
[19] R. Hiran, N. Carlsson, and P. Gill. Characterizing large-scale routing anomalies: A case study of the china telecom incident. In *Proc. PAM*, 2013.
[20] S. Iyer, A. Rowstron, and P. Druschel. Squirrel: a decentralized peer-to-peer Web cache. In *Proc. ACM PODC*, 2002.
[21] J. Karlin, S. Forrest, and J. Rexford. Pretty good BGP: Improving BGP by cautiously adopting routes. In *Proc. IEEE ICNP*, 2006.
[22] S. Katti, B. Krishnamurthy, and D. Katabi. Collaborating against common enemies. In *Proc. IMC*, 2005.
[23] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. PHAS: a prefix hijack alert system. In *Proc. USENIX Security Symp.*, 2006.
[24] M. Lepinski. BGPSEC protocol specification. IETF, 2013.
[25] M. Lepinski and S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480 (Informational), 2012.
[26] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and kc claffy. AS relationships, customer cones, and validation. In *Proc. IMC*, 2013.
[27] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An information plane for distributed services. In *Proc. OSDI*, 2006.
[28] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an accurate AS-level traceroute tool. In *Proc. ACM SIGCOMM*, 2003.
[29] P. Marchetta, W. de Donato, and A. Pescapé. Detecting third-party addresses in traceroute traces with IP timestamp option. In *Proc. PAM*, 2013.
[30] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *Proc. ACM SIGCOMM*, 2006.
[31] H. Ringberg, A. Soule, and M. Caesar. Evaluating the potential of collaborative anomaly detection. Technical report, 2008.
[32] Y. Shavitt, E. Shir, and U. Weinsberg. Near-deterministic inference of AS relationships. In *Proc. ConTEL*, 2009.



(a) Chord (mean) (b) IP-level (mean)

Figure 13: Number of Chord lookups and IP-level messages to resolve a query.

[33] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proc. ACM SIGCOMM*, 2001.
[34] Symantec. 2013 Internet Security Threat Report, 2013.
[35] Y. Zhang, R. Oliveira, Y. Wang, S. Su, B. Zhang, J. Bi, H. Zhang, and L. Zhang. A framework to quantify the pitfalls of using traceroute in AS-level topology measurement. *IEEE JSAC*, pages 1822–1836, 2011.
[36] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush. Ispy: detecting IP prefix hijacking on my own. *ACM CCR*, pages 327–338, 2008.
[37] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis. A light-weight distributed scheme for detecting IP prefix hijacks in real-time. In *Proc. ACM SIGCOMM*, 2007.
[38] C. V. Zhou, C. Leckie, and S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers and Security*, 29:124–140, 2010.

APPENDIX

A. OVERHEAD ANALYSIS

To evaluate the scalability of the routing overhead associated with longest prefix matching, we use a modified version of PlanetSim [15] to simulate the path the query message takes in alliances with varying sizes. The global repository was populated with all public routable prefixes that were available from the Cyclops project[4] on Sept. 23, 2012, and node identifiers were assigned at random for each simulation.

Figure 13(a) shows the number of Chord lookups for overlays with different numbers of alliance members. For each alliance size, we simulate the query path for one million random IP-address pairs and report the average. We also include a best fit curve of the form $c\log^{\alpha} N$, where α is a scale parameter. Figure 13(b) shows the corresponding statistics for the overall number of IP-level messages (without use of 1-hop optimization). Our polynomial fittings suggest that the power α is roughly 0.67 and 1.29, respectively, for the two metrics. The two metrics are identical for the case in which we use 1-hop optimization (equal to values in Figure 13(a)).

Finally, per-node storage overhead scales as $O(PH/N)$, and forwarding tables as $O(\log N)$ or $O(N)$, depending on whether 1-hop optimization is implemented. To put the per-node storage overhead into perspective, consider a scenario in which there are $P = 0.55$M prefixes, $H = 5$ holder nodes, and $N = 100$ alliance members. In this case, each node must on average store 25K prefixes, substantially less than the number of prefixes stored on a typical core router.

[4]Cyclops project, `http://cyclops.cs.ucla.edu/`, Sept. 2012. (This list included roughly 0.55 million prefixes.)