2011 IEEE Symposium on Security and Privacy

SCION: Scalability, Control, and Isolation On Next-Generation Networks

Xin Zhang, Hsu-Chun Hsiao, Geoffrey Hasker, Haowen Chan, Adrian Perrig and David G. Andersen
CyLab / Carnegie Mellon University

***Abstract*—We present the first Internet architecture designed to provide route control, failure isolation, and explicit trust information for end-to-end communications. SCION separates ASes into groups of independent routing sub-planes, called *trust domains*, which then interconnect to form complete routes. Trust domains provide natural isolation of routing failures and human misconfiguration, give endpoints strong control for both inbound and outbound traffic, provide meaningful and enforceable trust, and enable scalable routing updates with high path freshness. As a result, our architecture provides strong resilience and security properties as an intrinsic consequence of good design principles, avoiding piecemeal add-on protocols as security patches. Meanwhile, SCION only assumes that a few top-tier ISPs in the trust domain are trusted for providing reliable end-to-end communications, thus achieving a small Trusted Computing Base. Both our security analysis and evaluation results show that SCION naturally prevents numerous attacks and provides a high level of resilience, scalability, control, and isolation.**

I. Introduction

The Internet is the most geographically, administratively, and socially diverse distributed system ever invented. While today's Internet architecture admits some administrative diversity, such as by separating routing inside a domain (intra-AS routing) from global inter-domain routing, it falls short in handling the key challenges of security and isolation that arise in this intensely heterogeneous setting. As a result, we see surprisingly frequent incidents in which communication is interrupted by actions or actors far from the communicating entities. In addition to classical examples such as YouTube being globally disrupted by routing announcements from Pakistan [1], other issues surrounding the lack of resource control and isolation are not solved by existing proposals such as S-BGP [2]: the introduction of excessive routing churn [3]; traffic flooding; and even issues of global conflicts over naming and name resolution.

This paper proposes a clean-slate Internet architecture, SCION, that provides strong guarantees for failure isolation and route control in ways that map well to existing geographic, political, and legal boundaries. We show that strong control and isolation naturally leads to security and reliability without the use of high-overhead security mechanisms, while exposing to the endpoints diverse communication path sets that can support a wide spectrum of routing policies and path preferences (*path expressiveness*).

We introduce the notion of a hierarchy of *trust domains* whose members all share a common contractual, legal, cultural, geographical, or other basis for extending limited trust among each other. Examples may be a domain of U.S. commercial and educational institutions, ISPs that participate in the same peering point who share a common, binding legal contract on their behavior, or ISPs in the same state or country who are subject to the same laws and regulations. Using this abstraction, we provide the machinery to guarantee control-plane isolation: *Entities outside a trust domain cannot affect control-plane computation and communication within that trust domain.* For communication that must span trust domains, we provide the property that *the entities who can affect the communication are limited to a necessary and explicitly identified set of other trust domains.* We leave data-plane security as future work and thus do not consider denial of service attacks. In addition, the introduction of trust domains enables sources, transit ISPs, and destinations in SCION to agree *jointly* on which path to use. The architecture naturally controls routing information flow, and provides for explicit trust in path selection.

Through isolation and control, SCION enables expressive trust, i.e., *all the communicating endpoints can decide and control explicitly and precisely whom they need to trust for providing reliable communications*. Exposing such explicit trust information for end-to-end communication can eventually benefit network availability, because the endpoints can select more "trusted" communication paths with presumably more reliable data delivery; or at least, SCION holds the parties involved in the communications accountable for their misbehavior and failures.

Contributions. We design and analyze SCION, an Internet architecture emphasizing the principles of control, isolation and explicit trust. SCION enables route control for ISPs, senders and receivers at an appropriate level of granularity, balancing efficiency, expressiveness, policy compliance, and security. The isolation properties dramatically shrink the TCB and make explicit which entities communication relies upon. SCION offers strong security properties and demonstrates that the resulting routes widely mirror those in place under BGP today. We anticipate that the proposed architecture offers a useful design point for a next-generation Internet.

This research was supported by CyLab at Carnegie Mellon under grants DAAD19-02-1-0389, W911NF-09-1-0273 and W911NF-0710287 from the Army Research Office, and by NSF under awards CNS-1040801, CNS-1050224, ANI-0331653, and CNS-0520187. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, CyLab, or the U.S. Government or any of its agencies.

1081-6011/11 $26.00 © 2011 IEEE
DOI 10.1109/SP.2011.45

II. Limitations of Current Routing Design

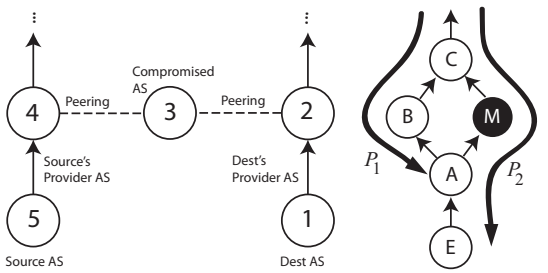
To motivate our new design, we first demonstrate four fundamental limitations of current inter-domain routing protocols. Through concrete examples and discussion, we show that recent popular inter-domain routing protocols [4]–[11], *even with their semantics perfectly secured (e.g., via S-BGP [2] like approaches)*, still lack several important security properties.

Limitation 1: Arbitrary information flow. Many current inter-domain routing designs use path-vector routing because it supports rich routing policies [12] and is more scalable compared to link state. In addition to BGP, recently proposed protocols such as MIRO [6], R-BGP [7], Routing Deflections [9], and ACR [10] also use path-vector route dissemination. These routing systems, however, give endpoints and ISPs little control over how their routing announcements propagate, which causes several security vulnerabilities. Specifically, once a node N announces its prefix, path, or pathlet to its neighbors, N has no control over the way in which its routing update is further propagated and paths are constructed for reaching N.

Figure 1(a) depicts an example scenario. Destination AS 1 is served by provider AS 2; the source AS 5 is likewise served by AS 4. An intermediate AS 3 peers with both providers; the providers do not peer with each other directly. If AS 3 wishes to control the route between the source and destination, it can forward the route it learns from AS 2 to AS 4 even if S-BGP is used. Because routes announced by peers are generally preferred over routes announced by providers, this will likely result in the destination using the AS-PATH $\{1, 2, 3, 4, 5\}$. Such a path violates the *valley-free* routing principle that a node should not provide transit service between two providers or peering neighbors. Such violations can cause routing *convergence* problems, where upon topology changes, the routing table at each node may not converge to the updated, correct routing paths [13]. Using conventional routing security measures, such as S-BGP, which only secures the strict semantics of path vector, it is impossible to distinguish this route from a legitimate route. Currently, the only practical method for dealing with such anomalies is to use hand-tuned ingress or egress filters to custom-configure the system, which can be error-prone and cause inconsistencies [14].

Figure 1(b) depicts a second example, where the endpoint AS E is the destination of traffic. AS E generates a route advertisement for its address prefix which is propagated through its provider A; this advertisement is further propagated into separate paths P_1 and P_2 going through B and M, respectively, and re-converging at AS C. Suppose AS C selects P_2 to re-advertise; then P_1 is discarded and all inbound traffic to E now must pass through the AS M which is less preferred by E.

In summary, routing systems with undirectional and unregulated flow of routing update dissemination can suffer from three problems: (i) "valley" paths can inhibit routing convergence; (ii) paths can traverse ISPs untrusted by the source node; and (iii) the routing system is generally subject to arbitrary blackhole and wormhole attacks. This unprincipled manner of path construction is a well-known source of persistent Internet route fluctuation [13].



(a) Valley-free violation. (b) Lack of inbound traffic control.

Fig. 1. Arbitrary information flow. Small arrows indicate customer-provider (downstream/upstream) relationships. Large arrows indicate constructed paths. In Figure 1(b), P_1 is preferred by the endpoint E but P_2 is the path that is used.

Limitation 2: No joint path selection between source and destination. The lack of joint path selection between the source and destination nodes prevents effective defenses against Denial-of-Service attacks. Traditional path-vector routing lets intermediate ASes select which routes to advertise to other peers and customers from the set of announcements they receive from their neighbors. Endpoints have no control over path construction. Newer proposals for multi-path routing [4]–[6], [9] recognize that the users of a path – the communicating endpoints – should have the final say over a route's acceptability. These proposals let the *source* select from a set of diverse paths, but they do not similarly empower the destination to control its inbound traffic, as illustrated in Figure 1(b). Consequently, the destination has little inbound traffic control to avoid using particular untrusted nodes for its own communication.

Limitation 3: Lack of routing isolation. A central tenet of current inter-domain routing architectures is reachability, where a routing announcement from any AS can potentially be propagated throughout the entire Internet. In other words, most (if not all) ASes are in the same routing dissemination domain. For example, in addition to the aforementioned multipath routing protocols, NIRA [15] organizes all the ASes in one tree-based routing domain, and Landmark routing [16] also makes the routing "landmarks" available throughout the network. While such global visibility helps achieve global reachability, it also enables individual malicious ASes to easily launch attacks affecting the *entire* Internet. For example, two distant colluding ASes can announce a (non-existing) wormhole link between each other to create a (bogus) short path, which can be seen potentially by the entire Internet and thus attract traffic.

Limitation 4: Lack of route freshness. An adversary who can delay or drop messages can force traffic to continue to use an older path p with obsolete state. Because routing updates from each AS have global scope, current inter-domain routing protocols send only incremental routing updates after route changes to achieve scalability. Unfortunately, this incremental manner of routing updates sacrifices route freshness, as the loss of updates concerning a path p (such as path withdrawal messages) can prevent other ASes from knowing that path p has changed. Consider the example in Figure 1(b), where the

AS PATH $\{C, M, A, E\}$ is active to reach destination E. Suppose that A withdraws the path $\{A, E\}$, but the malicious AS M intentionally suppresses this withdrawal message from C. Consequently, the same AS PATH $\{C, M, A, E\}$ still remains active, because in path vector routing B only withdraws a *path* $\{B, A, E\}$ instead of a specific link, which does not invalidate the path through M.

III. SCION Overview

SCION has three grounding principles: domain-based *isolation*, mutually *controllable* path selection by both the endpoints and intermediate ISPs, and explicit trust for end-to-end communication, as Section III-A details. These principles provide a framework within which SCION achieves resilience to routing attacks. The rest of the section provides an overview of the SCION architecture.

A. Design Principles

Principle 1: Domain-based isolation – Dividing the routing control plane into independent domains. Isolation among independent domains protects routing in one domain from malicious activities and routing churn in other domains. This benefits both security and scalability while retaining reachability and path diversity across domains. For example, SCION enables frequent routing updates to periodically refresh path state, so that each AD always maintains a fresh (and accurate) network topology for efficient routing decisions.

Principle 2: Mutually controllable path selection – Joint path selection between source and destination. SCION greatly increases *both the source and destination's ability* to affect, select and control the construction of the routes to and from themselves, while still respecting intermediate ISPs' routing policies.

Principle 3: Explicit trust and small TCB for end-to-end communication. By segregating mutually distrustful entities into different trust domains, each trust domain can choose a coherent root of trust (e.g., a few tier-1 ISPs) for bootstrapping trust among ADs in the same trust domain. As a result, an endpoint E knows and is able to choose explicitly whom to trust for achieving reliable end-to-end communication, while untrusted ADs in other trust domains cannot affect the path discovery and route computation of E. Consequently, an entity only has to trust a small subset of the network thus achieving a small TCB for end-to-end communication.

B. Hierarchical Decomposition

Our architecture defines the Autonomous Domain (AD) as the atomic failure unit, representing both ISPs (or transit ADs) and endpoint ADs. Large ISPs would be split into multiple ADs, based on their topology of separately administered domains. SCION divides the ADs in the Internet into a hierarchy of *trust domains*, or TDs, as shown in Figure 2, used to provide the domain-based isolation property. A TD is a set of ADs that agree on a coherent root of trust and have mutual accountability and enforceability for route computation under a common regulatory framework.

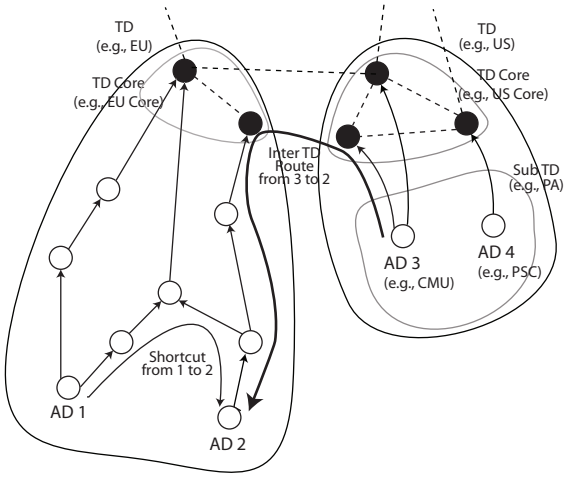


Fig. 2. Trust domain architecture. Black nodes are ADs in the TD Core. Arrows indicate customer-provider relationships. Dashed lines indicate peering relationships.

Each TD has a *TD Core*, a set of designated ADs forming a mutually reachable clique that interfaces with other TDs. ADs in the TD core naturally serve as the egress/ingress ADs of the corresponding TD. In the current Internet, the top-tier ISPs would constitute the TD Core.

We envision the effort to establish a TD to closely mirror that of starting a certification authority, and the number of top-level TDs to be limited (e.g., up to a few hundred) which map to real-world political or cultural groups. Section IV presents a detailed description of a TD.

C. Routing, Lookup, and Forwarding

All ADs in SCION know a set of paths to reach the TD Core in their trust domain for establishing communication with other endpoint ADs. Specifically, for an AD N that is not in the TD Core, we call the paths for sending packets from N to the TD Core **up-paths** of N, and the paths for sending packets from the TD Core to N the **down-paths** of N, which are not necessarily different from the up-paths.

The down-paths of each AD N are available to other ADs via a lookup service, and are used by other ADs to reach N. To communicate with a destination AD D in another TD, the source AD S selects a subset of its up-paths to reach the *top-level* TD containing S for sending data to D, and can pick an independent subset of the down-paths for receiving data from D. In this way, ADs retain control over the paths for both outgoing and incoming data within their own TDs.

In the following, we first sketch routing, name lookup, and forwarding between two endpoint ADs in the same TD, and then briefly explain how cross-domain communication is enabled.

Path construction. In SCION, ADs use a set of up-paths/down-paths to send/receive packets to/from the TD Cores. We generally refer to these up-/down-paths as *paths*, which are constructed similar to path vector as follows. The ADs in the TD Core first transmit *one-hop* paths starting from

the core to their 1-hop customer ADs via *path construction beacons*. These customer ADs then add themselves to the path and propagate the received paths to their customers and peers, and so on. Each *endpoint AD* then selects among all the paths received from each provider or peer to form its own k (ideally maximally disjoint) up-paths for reaching the TD Core and down-paths for receiving packets from the TD Core.

Lookup. The endpoint ADs publish the selected down-paths on the TD's Path Server, a service located in the TD Core, queried by local and foreign ADs for routing information. SCION employs Accountable IP (AIP) [17] for host and AD addressing, where each address represents a public key. The TD Core signs a TD membership certificate for each AD address. A name lookup takes as input a human readable name of the destination, and returns both the AIP address of the destination host and AD, and the AD-level down-paths published by the destination AD.

Path selection. To form a complete end-to-end communication path to reach a destination, the source AD first chooses one of its up-paths to reach the TD Core and queries the destination's down-paths via name or address lookup. The source AD then selects one of the queried destination's down-paths to construct a complete end-to-end path. For simplicity, the gateway in an endpoint AD makes the default path selection decision on behalf of the hosts in that AD, while a host can also negotiate with its provider AD to support customized path selection policies.

Route joining. Before naively combining one of the source AD's up-paths with one of the destination AD's down-paths, the source AD searches the paths for common ancestor ADs, to find a "shortcut" path without passing through the TD core. Figure 2 shows an example shortcut that can be found between AD1 and AD2.

Forwarding. Once a source AD constructs a complete end-to-end communication path, the source AD embeds in each packet certain "opaque fields" created by the transit ADs during path construction, which encode the forwarding path information as ingress/egress points at each transit AD in the end-to-end path. Within each AD, any internal routing protocol can be used to find a path from an ingress point to an egress point. The destination can simply reverse the embedded path or query the source AD's Path Sever for alternative paths to reach the source. Hence in SCION, packet forwarding between ADs eliminates the need of routing and forwarding tables.

TD-level routing. When communication crosses domains (e.g., a source wants to reach a destination in another TD), TD-level routing enables each TD to determine the routes to other TDs. TD-level routing takes place using pre-negotiated, human-configured routes or source routing to enable explicit path control. Given the envisioned small and stable topology of top-level TDs (e.g., around one hundred TDs), scalability and routing security are no longer major concerns for Inter-TD routing.

D. Policy Enforcement

In SCION, the stakeholders of end-to-end communication impose their policy decisions in three stages:

1) Transit ADs apply their routing policies when deciding which paths to propagate via path construction beacons.
2) Destination ADs apply policy in their selection of k down-paths to publish at the TD's Path Server.
3) Source ADs apply policy to select an up-path to the TD Core and one of the down-paths retrieved from the Path Server to reach the destination.

E. Small Trusted Computing Base (TCB)

In SCION, TDs provide natural boundaries for failure isolation and domains for strong routing control. SCION assumes only that the *TD Cores are trusted* by the ADs in the *same* TD, but does not assume that ADs in the same TD, nor a remote TD Core is trusted. Consequently, the TCB for the end-to-end communication between two endpoints consists of the TD Core and only ADs in the corresponding up-paths and down-paths, whereas in the current Internet architecture the end-to-end communication can be potentially affected by any node in the network. Since ADs in one TD share the same contractual or cultural goals, reach the same business or technical agreements, and are subject to the same laws and regulations, the activities of ADs in the same TD are held *accountable* for their route computation and deviations are enforceable because every TD represents a uniform legal environment. Furthermore, the TD Core in SCION serves as the root of trust to bootstrap trust and enforce security policies among ADs in that TD.

IV. Anatomy of a Trust Domain

A trust domain (TD) is the fundamental unit of trust in the SCION architecture. TDs are *communities* of network entities held together by *enforceable rules* such as contracts, shared legislative and judicial frameworks, or physical locality. Given these aggregates, the fundamental goal of the architecture is to enforce isolation between TDs while providing interconnection. Each trust domain can be considered an independent networking plane shielded against the influence of external entities. The global goal of the architecture is to allow any endpoint to explicitly specify which set of these networking planes it wishes to use and facilitate a connection based on these requirements.

Figure 3 presents the architecture of a TD. Conceptually, a TD is composed of a contiguous set of ADs along with their explicitly marked customer-provider relationships. A specially designated set of *tier-1* ADs, called the TD Core ADs, represents the top level of the AD hierarchy: this set contains the entities that perform several authoritative functions of the trust domain, e.g., managing the certificates and public/private keys for that TD. The set of TD Core ADs must be connected and mutually reachable in the AD-graph (e.g., routing between any ingress and egress points of the TD Core, and reaching ADs that implement the Path Servers for name lookup). Since most TD Core ADs in the current Internet are densely peered,

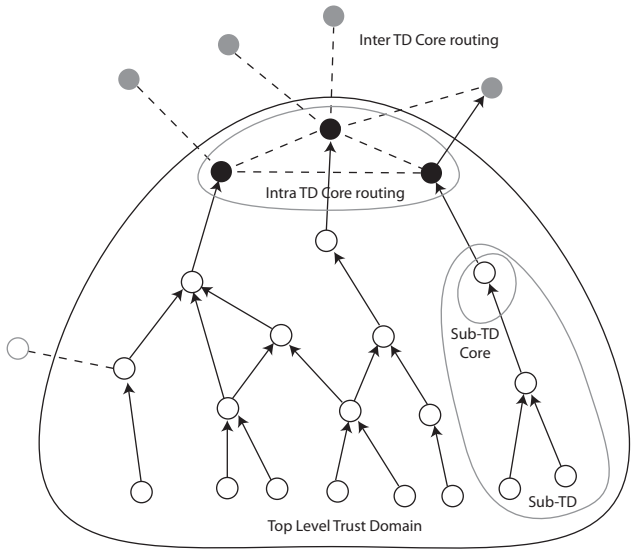


Fig. 3. A Top-level trust domain. Black nodes are ADs in the TD Core. Arrows indicate customer-provider relationships. Dashed lines indicate peering relationships.

routing in this topology should be simple. Nevertheless, to enable path choice (i.e., to avoid routing traffic through an untrusted TD), we propose that a link-state routing protocol for topology discovery be used in conjunction with source routing between TD Core ADs.

A trust domain can be a top-level trust domain (TLTD), or a sub-TD. A sub-TD resides completely within a TLTD and may contain other sub-TDs. No other TD fully contains a top-level TD, although its member set may overlap partially with other TDs. As mentioned before, we anticipate that relatively few top-level TDs will form (up to a few hundreds), with each TD corresponding to a large, globally identifiable real-world group (such as a country, or a well-known international organization).

The TD Core ADs in the top-level TDs facilitate interconnection between top-level TDs using the Inter-TLTD routing protocol. Since this topology is extremely small and densely connected (the majority of routes should not need to traverse more than 2 TDs), most of the routes are static and can be directly configured. When automatic route discovery is needed we assume that TD-level routing policy (e.g., which TD to use to reach a distant TD) is agreed-upon among the TD Core ADs beforehand. To facilitate the Inter-TLTD routing protocol, the TD Core ADs engage in a protocol to discover their mutual interfaces to other TLTDs in a manner similar to IGP; since there are only a few of these TD Core ADs per TLTD, each TD Core AD can simply keep a table of what TLTDs are reachable from each of its fellow TD Core ADs.

A. Trust Domain Membership

Identifiable organizations (a government, an industry consortium, etc.) administer trust domains.

When a new AD wishes to join a TD, the TD Core verifies that the AD meets the requirements for membership in the TD (for example, a country-based TD may require that the corresponding ISPs be registered and headquartered within a given country). Then the TD authority determines the topological relationship of the joining AD with respect to the current TD. To join an existing trust domain, the new AD should either: 1) have one provider already inside this TD, or 2) be capable of being a core AD in this TD. To meet the second requirement, the new AD must be able to *directly* reach other core ADs, as well as satisfy a subjective assessment of the AD's connectedness: this requirement mirrors customer / traffic requirements for peering.

When an AD establishes a new connection with an AD in a different trust domain, it must join one or more of the TD associations of the provider in order to access the relevant sets of paths of that provider. The exact TD assignment is dependent on the terms of the service and contingent on whether the child AD can satisfy the conditions of joining the new TDs.

B. Management and Trust Bootstrapping

The TD Core ADs administer each TD. For simplicity we assume that a single entity performs this function, but a distributed approach likely prevails in practice. Each top-level TD has a fixed human readable identifier as well as a public/private keypair K_{TDC}/K_{TDC}^{-1}. In practice, each AD in the TD Core can possess a public/private key pair, and a threshold number of ADs are required to generate a valid TD Core signature. Due to the high level of visibility of the top-level trust domains, we assume that bootstrapping the well-known TD Core public key K_{TDC} onto the relevant principals (specifically, the member ADs in that TD as well as other top-level TD authorities) occurs securely. For example, service providers could pass on the public key to their customers. The TD Core then operates a PKI CA for the member ADs of that TD, signing certificates of membership binding ISP identification and AD numbers to ADs and their respective public keys. The TD CA can either be implemented as a dedicated server, or split among multiple ADs in the TD Core.

C. Subsidiary Trust Domains

A top-level TD may contain subsidiary trust domains (or *sub-TDs*). Figure 3 depicts a sub TD inside the main top-level TD. Sub-TDs allow finer-grained trust domain selection (for example, the armed forces of a country may operate a sub-TD within its own country's TD, to support a higher level of assurance than civilian ISPs). A sub-TD's internal structure mirrors a top-level TD, with its own full mesh of ADs as the sub-TD Core, its own name lookup servers, etc. An endpoint AD also maintains a set of up-paths and down-paths per sub-TD Core containing that endpoint. The sub-TD's absence from the inter TLTD routing protocol provides the only distinction between a sub-TD and a top-level TD.

D. Benefits of Using Trust Domains

We end this section with a list of intrinsic benefits of building the inter-domain routing architecture based on the notion of TDs.

Security against attacks. The strong isolation and control that SCION's TDs provide naturally eliminates multiple long-standing attacks. For example, TD-based isolation intrinsically eliminates malicious messages and information from other TDs. ADs within the same TD have enforceable accountability for their route announcements and computation, because the ADs are regulated by the same legal framework. Furthermore, outbound traffic control enables the source ADs to bypass malicious or untrusted transit ADs, and inbound traffic control enables the destination ADs to efficiently stop, shape, or regulate unwanted incoming traffic.

Resilience against misconfiguration. The intrinsic isolation provided by the division of TDs achieves in-depth resilience to human error, a prevailing reason for current routing system outages [18]. First, the simplicity and convenience of needing only to design a robust inter-domain routing architecture, likely to remain relatively stable over time, helps reduce human configuration errors as opposed to "ad hoc" engineering hacks prevalent in practice. Second, the TD structure mitigates the damage to its member ADs and those attempting to reach them.

Elimination of a single point of trust/failure. The existence of multiple TDs eliminates the need for a single authority for the entire Internet, which causes deployment issues and a single point of failure. For example, DNSSEC currently requires a single root of trust for the entire Internet, whereas in SCION each TD maintains its own root-of-trust authority.

Scalability. By scoping the route dissemination and computation within each TD independently, SCION also achieves routing scalability. In each TD, only the TD Core originates routing messages (the path construction beacons), which are only propagated within the TD. In contrast in path vector or link state, every node in the network can generate routing updates, which are disseminated throughout the network. Such routing scalability enables the use of proactive, frequent path construction beacons by the TD Core, ensuring that each AD can learn fresh path state within the TD to address the route freshness problem stated in Section II.

V. Path Construction

In this section we describe how each AD learns of its AD-level paths to the TD Core through periodic *path construction beacons* containing routing information.

Up-path and down-path selection. Upstream paths, or "up-paths" are a set of paths that endpoint ADs select from path construction beacons for reaching the TD Core. To support multi-path routing, the protocol enables all ADs to receive multiple distinct policy-compliant AD-level paths to reach the TD Core. To keep route lookup overhead practical, we restrict the number of paths that each endpoint AD maintains in its reachability record to at most k per TD (an endpoint may be contained in multiple TDs). In facilitating the construction of these up-paths, the upstream ADs need not exhaustively enumerate all possible paths to the TD Core, but must simply provide a sufficient number of alternatives. Downstream paths, or "down-paths", are a set of k paths an endpoint AD selects from path construction beacons to upload to the TD's path lookup server for others to reach that endpoint AD. An endpoint AD may select a set of different down-paths than the up-paths to implement independent inbound and outbound traffic control.

Overview of path construction. Path construction enables endpoint ADs to find their up-paths and down-paths, or *paths* in general between the endpoint ADs and the TDCore. Construction begins with each TD Core AD initiating a path construction beacon in every time period. Each AD passes along a path construction beacon to each customer and peer in the same TD, appending additional information. A path construction beacon, denoted by $U = \{(p, \mathbb{G})\}$, is a set of pairs $(p, \mathbb{G})$ with each pair corresponding to an announced path. p and $\mathbb{G}$ are as follows:

- a set of links p within a TD assembling a path for reaching the TD Core, where each link in path p is timestamped and authenticated; and
- a set of global information $\mathbb{G}$ pertaining to all the links in path p, such as the timestamp TS when the TD Core initiated the path construction beacon and the TD identity TD. The TD Core also signs $\mathbb{G}$ with K^{-1}_{TDC}, and $\mathbb{G}$ will not be changed by ADs during path construction.

Each link in p also contains an *opaque field*, which includes short cryptographic markings, similar to stateless network capabilities [19], generated by the AD owning that link to enable efficient forwarding control.

Often, routing at the AD granularity is insufficient. For example, consider a large carrier with continental reach but only a single AD number. A route passing through the AD could be entirely local, or it could cross the entire continent. Hence, SCION paths specify a sequence of ingress/egress interfaces at each upstream AD for reaching the TD Core. At each hop during path construction, the local AD first gathers path information from the path construction beacons received from neighbors, selects a subset of paths to be further announced, and appends additional information to the selected paths to form new path construction beacons. The additional information added includes an ingress/egress interface pair *per path* to specify how traffic should enter and exit the local AD *along that path*. Each ingress or egress interface corresponds to a single neighboring AD to uniquely define the preceding or following AD along with the TD scope (i.e., in which TD can this link be used); this enables bi-directional forwarding. Because the beacon timestamp TS and TD scope TD in $\mathbb{G}$ in a path construction beacon are not changed by ADs, we focus on p.

The remainder of this section details our path construction protocol which supports *peering links* and *fine-grained routing based on ingress/egress points*. Table I summarizes the notation used in the following protocol description. We first specify the format of a path construction beacon, and then walk through the path construction process using an example topology.

A. Format of a Path Construction Beacon

Suppose AD_{i-1} passes a path construction beacon to its neighbor AD_i (e.g., a customer or a peer) along a path p. Then, a link $p \in p$ in the path construction beacon, which AD_i further announces along p, contains four fields $I_p(i)$, $T_p(i)$, $O_p(i)$, and $\Sigma_p(i)$ as detailed below. The path construction beacons are TD-scoped and only propagated within the TD from which they are originated.

1) Interface field $I_p(i)$: This field contains the path received from the previous hop, appended with the local link (ingress/egress interface pair). That is:

$$I_p(i) = I_p(i-1)||ingress_p(i)||egress_p(i) \tag{1}$$

where $ingress_p(i)$ and $egress_p(i)$ denote the interfaces that connect AD_i with the previous hop and the next hop during path construction, respectively. We enforce that each AD_i labels its interfaces TD-specific, if AD_i joins multiple TDs; so that from an interface label AD_i knows for which TD that interface is used.

2) Timing field $T_p(i)$: The timing information for each link contains an expiration time depending on the AD's internal policy. The introduction of link expiration time requires endpoint ADs to periodically select new, updated sets of paths, thus achieving route freshness.

3) Opaque field $O_p(i)$: For packet forwarding, the opaque field will be embedded into data packets such that ADs on the path can efficiently check (i) whether the incoming traffic is allowed and enters from the correct AD at the correct ingress point, and (ii) the corresponding egress interface to the next-hop AD. Hence, $O_p(i)$ contains the ingress/egress interfaces at AD_i for the announced path p, and a short crypotographic Message Authentication Code (MAC) by which an AD can verify if the opaque field embedded in the data packet is authentic. Specifically:

$$\begin{aligned} O_p(i) =& ingress_p(i)\|egress_p(i) \\ & \text{MAC}_{K_i}(ingress_p(i)\|egress_p(i)\|O_p(i-1)) \end{aligned} \tag{2}$$

The MAC portion is computed using a secret key K_i known only to AD_i, and is essentially a *data plane capability* used to remind AD_i of its own decision that p is an approved path when used to carry data packets. Note that since these capability MACs are only verified by the issuing AD (AD_i), it requires no time synchronization with other ADs as long as the time synchronization within the routers of each AD are sufficient to enforce its individual policy regarding route announcement timeouts. An opaque field $O_p(i)$ will only be checked by the issuing AD_i during forwarding, while its semantics remain "opaque" to other transit and endpoint ADs. Hence, an AD can put a wide range of information in the opaque field to support flexible routing policies, as we discuss in Section XI.

In addition to $O_p(i)$, AD_i also includes in the path construction beacon the opaque fields received from preceding ADs in p, denoted by $\{O_p(j)\}_j$, where AD_j is an ancestor of AD_i in p. In this way, an endpoint AD can receive all the opaque fields by its upstream ADs, which can later be used for efficient forwarding as we show in Section VIII.

TABLE I
NOTATION.

	Notation	Meaning
AD cert	$\text{Cert}_{i\in TD}$	TD certifies AD_i is in TD
Sub-TD Cert	$\text{Cert}_{TD1\subseteq TD}$	TD certifies TD1 is its sub-TD
TD Core Cert	$\text{Cert}_{TD\to i}$	TD certifies AD_i is a TD Core AD
Signature	$\text{Sign}_i(X)$	AD_i signs X with private key K_i^{-1}

An AD can use different secret keys for generating MACs for different beacon timestamps TS. In practice, an AD may employ a "grace period" within which an expired link can still be used and the corresponding MAC is still valid. Consequently, at any point of time, multiple usable timestamps can co-exist thus requiring an AD to maintain a small key table storing the concurrent secret keys for MAC generation. Upon receiving a data packet, the AD will use the beacon timestamp TS along with the default expiration time of the paths within that AD to retrieve the key for MAC verification. At any point in time, numerous symmetric keys K_j corresponding to different expiration times will be valid. Thus, routers will need different keys for verifying the MAC of opaque fields, depending on the expiration time which in turn can be derived from the path timestamp TS which is included in the packet header. The different keys K_j can be derived through the use of a PRF F that is keyed with a secret key K_{AD} known by all routers in the AD and computed over a key index j: $K_j = F_{K_{AD}}(j)$. For the function F we suggest using AES in ECB mode, assuming that the key size for keys K_j is at most 128 bits. As a consequence, only the secret symmetric key K_{AD} needs to be distributed among all routers in the AD.

4) Signatures $\Sigma_p(i)$. AD_i signs its local link in p as follows:

$$\Sigma_p(i) = \text{Cert}_{i\in TD}\|\text{Sign}_i(I_p(i)\|T_p(i)\|O_p(i)\|\Sigma_p(i-1)) \tag{3}$$

where $\text{Cert}_{i\in TD}$ is a membership certificate authenticating AD_i as a member of TD, and the AD number can be extracted from this certificate. Note that the signatures are constructed in an *onion* fashion, where each signature signs all previous path information.

If AD_i is in the TD Core, Σ_i also includes an additional certificate $\text{Cert}_{TD\to i}$ signed by the TD authority authenticating AD_i as a TD Core AD:

$$\begin{aligned} \Sigma_p(i) =& \text{Cert}_{TD\to i}\|\text{Cert}_{i\in TD}\| \\ & \text{Sign}_i(I_p(i)\|T_p(i)\|Q_p(i)\|\Sigma_p(i-1)) \end{aligned} \tag{4}$$

B. Supporting Peering Links

To identify shortcuts across peering links (as described below), each pair of peering ADs (i, h) also exchanges a peering certifier $Q_{i,h}$ that AD_i inserts to p with the following

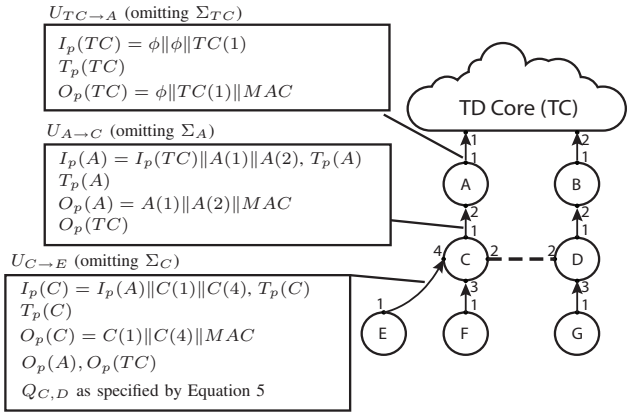


Fig. 4. Path construction beacon format along path $p = \langle TC, A, C, E \rangle$. The link between C and D is a peering link; other links are customer-provider links. The symbol ϕ denotes an empty field, and $AD_i(n)$ denotes the interface labeled n of AD_i (e.g., $TC(1)$ represents the interface labeled 1 of the TC). Furthermore, MAC refers to the MAC constructed per Equation 2.

fields:

$$\begin{aligned} &I_{i,h}(i) = ingress_{i,h}(i) \| egress_{i,h}(i) \| TD_h \| AID_h, \\ &T_{i,h}(i), \\ &O_{i,h}(i) = ingress_{i,h}(i) \| egress_{i,h}(i) \| \\ &\qquad \text{MAC}_{K_h}(ingress_{i,h}(i) \| egress_{i,h}(i)), \\ &\Sigma_{i,h}(i) = \text{Cert}_{i \in TD} \| \text{Sign}_i(I_{i,h}(i) \| T_{i,h}(i) \| O_{i,h}(i) \| O_p(i)) \end{aligned} \tag{5}$$

where $ingress_{i,h}(i)$ and $egress_{i,h}(i)$ denote the ingress and egress interfaces of AD_i for traversing this peering link along path p; TD_h and AID_h denote the TD ID and AD number of the peer AD_h; and $T_{h,i}$ is the expiration time for that peering link.

C. Path Construction Process

We detail the path construction process given the above beacon format. Figure 4 provides a concrete example.

The TD Core periodically disseminates path construction beacons to the immediate neighboring ADs, e.g., every 15 seconds. Note that there is no "previous hop" for the TD Core for path construction beacon dissemination, the ingress interface and previous-hop opaque field are empty (as shown by ϕ in Figure 4). As an intermediate AD receives paths from its neighbors, it disseminates them to subsequent neighbors in the same TD. Suppose an intermediate AD (AD_i) receives a set of paths from its upstream providers. It checks the signatures on each of the paths, and discards any ill-formed or unauthenticated paths with bad signatures.

For each downstream AD (AD_j), the parent AD (AD_i) then chooses a (preferably maximally disjoint) path set of m paths $p_1, \ldots, p_m$, where $m \leq k$. Each of these paths necessarily originates from the TD Core and terminates at the parent AD, AD_i. For each path p, AD_i considers the set of its peering links that it will support for downstream customers and attaches this information into the path. It then updates $I_p(i), T_p(i)$, and $O_p(i)$ per Equations 1 and 2, respectively.

This process continues until each AD has obtained a set of paths originating from the TD Core and terminating at itself, where each link in the path is authenticated by each ancestor AD. Each opaque field essentially represents a network capability given from an ancestor provider to a customer; when these routes are used, the ancestor AD will check the corresponding MAC to ensure that the provided path corresponds to a path that it supports for that customer.

Once an endpoint AD receives the path construction beacon from its providers, the endpoint AD selects up to k up-paths and k down-paths, and signs the entire set of down-paths and their authenticators, and uploads the down-paths to a path lookup server provided by the trust domain to enable reachability. To send a packet, the source AD queries the path lookup server to find the down-paths associated with the destination AD and thus splice together an end-to-end route to reach the destination. Sections VI and VII describe this process.

VI. Lookup

SCION naturally enables the design of lookup protocols with explicit scoped trust, in accordance with the same foundational principles of the routing design. Trust information is made explicit by allowing endhosts to scope name resolution, i.e., to restrict the set of TDs responding to or involved in the resolution process. As lookup protocols can be orthogonal to the control-plane protocols addressed in this paper, in the remainder of this section we only specify a basic lookup mechanism for illustration purposes.

The SCION lookup process consists of two stages. i) An Address Server translates a human-readable label and a TD identifier into one or more cryptographic endpoint identifiers (EIDs) with their respective AD and TD memberships. Address Servers can support queries on EIDs as well. ii) A Path Server takes such AD and TD-membership information of an endhost and returns the destination AD's k down-paths. The replies from Address Servers and Path Servers are signed by the destination TD Core's private key K_{TDC}^{-1} to prevent attacks akin to DNS poisoning.

We assume that knowing the identity of the TD Core implies possession of the public key of that TD Core; this is similar to the assumption of, e.g., browsers having ICANN's root public key for DNSSEC. Also, to certify the mapping between the identifier (human-readable label or EID) and the address, each TD Core effectively maintains its own autonomous endpoint identifier space and signs a certificate authenticating the correctness of the name lookup. Each self-certifying AD identifier (or AID), named using AIP [17], comes with a co-signed certificate by the cores of TDs containing that AD, attesting to the membership of the AID in each of its respective TDs. Each AID:EID pair also contains a certificate signed by the corresponding AID's private key indicating that the EID is part of that AID.

A. Address Resolution Service

Context-aware address resolution. As highlighted above, to look up a path the source queries the local Address

Server with the destination identifier. The destination identifier consists of a human-readable label, such as a DNS name, and a TD identifier, which defaults to that of the local TD. The TD identifier indicates the context (i.e., TD) in which the label can be correctly interpreted. The Address Server returns an identifier indicating which AD(s) (or AIDs) serves an endpoint associated with the label, along with which sub-TD(s) contains the AID. Should the local Address Server not contain the name, the server will query other TDs' Address Servers on the user's behalf. Furthermore, the design settles disputes by resolving non-TD specified addresses at the local domain, where presumably an enforceable dispute resolution process exists.

Namespace. The namespace can be flat or hierarchical. For now, we assume that the service is structured similarly to DNS: there is one canonical root server associated with each trust domain, which can delegate the lookup to a number of sub-servers in possible other sub TDs, until a query is resolved. While SCION remains agnostic to the exact scheme for human readable naming, the example of DNS provides the most accessible example. Consider the domains ABC.us and ABC.cn, residing within the US and CN TD, respectively. A user within the US TD will query the local Address Server and receive address information to ABC within the US TD. Should the user in the US TD request ABC.cn (i.e., scoped "cn"), the US Address Server queries the CN Address Server, and returns this information to the user.

Address Server setup. Every TD provides an Address Server within its TD Core, accessible at a default address, which will resolve the name locally if possible, or query the appropriate external Address Server should the user specify a different TD. To ensure that Address Servers store the latest records, Address Servers should update their address database whenever an endhost joins or leaves. The update information can be provided by the endhost itself or the ADs involved. For example, an endhost x moves from AD_1 in TD_1 to AD_2 in TD_2 would trigger an address update event to add an entry (AD_2:EID_x)($AD_2 \subseteq TD_2$) to the Address Server in TD_2 and remove (AD_1:EID_x)($AD_1 \subseteq TD_1$) from TD_1's server (or add a redirection pointer indicating that x is temporarily move to AD_2 such as the mobile IP solution).

Routing to Address Server. The name resolution query is routed using the TD identifier. If the querying source is contained in the targeted TD (TD_t), the source sends this query directly to TD_t via one of its up-paths in this TD; otherwise, the query is sent to the top-level TD, which then resolves the query (possibly by querying sub-TDs or other top-level TDs) and returns the response via the originating up-path.

An example. The name resolution service takes as an input a human readable name: N_E (e.g., "Ford") and a TD Identifier C (e.g., "US"), and outputs a list of (self-certifying) AIDs and endpoint identifiers (EID): each result is an AID:EID pair constructed in a way similar to that of AIP [17]. Optionally associated with each AID:EID pair is a hierarchical nesting of trust domains (e.g., local, regional, and continental trust domains) that can be used to delegate the reachability functionality to sub-TDs. Specifically, a query on N_E at TD C should produce a record signed by TD C's private key $K^{-1}_{TD_C}$:

N_E in TD C resolves to:

AD:endhost	TD memberships
$AID_1 : EID_1$	$AID_1 \in TD_{1,1} \subseteq TD_{1,2} \subseteq TLD_C$
$AID_1 : EID_2$	$AID_1 \in TD_{1,1} \subseteq TD_{1,2} \subseteq TLD_C$
$AID_2 : EID_3$	$AID_2 \in TLD_C$

Here, the lookup of N_E under TD C returned three records: two endpoints EID_1, EID_2 in the same AD (AID_1) and another in a second AD, AID_2. The trust domain memberships of each AD are indicated: for example, AID_1 is contained in $TD_{1,1}$ which is a sub-domain of $TD_{1,2}$ which is a subdomain of the top level domain TLD_C. AID_2 is simply indicated as a member of the top level domain TLD_C.

B. k-Path Resolution Service

At this point the source possesses the AID:EID of the label that was looked up, as well as possibly a hierarchy of nested trust domains that contain this AID, but not a route to that AD:EID. The source now issues a route lookup query to the respective trust domains as appropriate (e.g., if the AID is contained in TD1 which is contained in TD2, and the source knows how to reach the Path Servers of TD1, it can contact TD1 directly). In the following description we assume that the source has no advance information and can only reach its own top-level TD using one of the AD's up-paths. The path resolution query from the source first goes to the top level TD ingress point, which may then query one of the top-level Path Servers to see if the down-paths have been uploaded. If not, or if the top level trust domain prefers not to resolve individual AD paths, it may also delegate the lookup based on the TD containment information provided in the address query. Eventually, a trust domain is found that contains the destination AD and whose Path Servers have a fresh copy of the k up-paths of the AID.

Path Server update. Through SCION's periodic path construction beacons, every endpoint AD obtains a fresh set of k paths in its TD. Whenever an AD selects a different set of paths due to policy changes or path failures, the AD actively updates its new k-path information to the Path Server. Consequently this enables *in-bound traffic control*, another way for the endpoints' routing policies to be reflected. A destination AD can attach *distinct* sets of k-paths depending on the identity of the source endpoints that are performing the query. For example, if a query is originating from outside the local TD, the destination AD can instruct the Path Server to only provide paths that pass through a set of specific high-security gateway ADs; whereas if the query originates inside the TD then a more general and efficient set of paths can be served.

Trust-scoped path resolution. A *trust scoped path query* enforces that a given path computation should only involve (and be restricted to paths using) ADs within a specific set of trust domains. For example, if a query is scoped to within top

level domain C, any AD that is not either explicitly specified by the querier, or in C, should not be able to directly affect the communication or the results of the query. For top-level scoped reachability queries, the implementation is straightforward. The querier sets a flag in its query indicating that this is a trust-scoped query restricted to the top level trust domain C. Then, when a Path Server is queried for paths, it simply withholds any paths that are not scoped for C. If the Path Server needs to follow a delegation path (e.g., querying the Path Server of a sub-TD), the Path Server ensures that the delegation path never exits C (in terms of forwarders or communication principals). This effectively restricts the set of participants in the path computation strictly to members of C. A label lookup can be scoped to a top level domain in a similar way.

Subtleties arise if a narrower scoping is desired. For example, suppose a source issues a path query $(AID_1 : EID_1)(AID_1 \in TD_{1,1} \subseteq TD_{1,2} \subseteq TLD_C)$, with the trust scoping constraint of $TD_{1,2}$ (where $TD_{1,2} \subseteq TLD_C$). Since $TD_{1,2}$ may not be a top level TD, and may not participate in the inter-TD top level routing layer, this implies that the path query is expected to traverse TLD_C but that the ADs in TLD_C (and, indeed the authority of TLD_C itself) may not be trusted. In such a situation, the source is required to designate a particular explicitly source-selected route to reach $TD_{1,2}$. The ADs in this route are allowed to drop the packet if it violates their routing policy but are expected to strictly follow the semantics of the routing otherwise. It is the responsibility of the source to discover a trusted route to reach the trusted TD $TD_{1,2}$; this can involve a separate path query (possibly, scoped to a group which contains only the high-level members of the top level domain TLD_C), or the route can be hand-configured based on source preferences.

VII. Route Joining

Route construction (Section V) allows an endpoint AD to determine its k up-paths to its core; route lookup (Section VI) allows a source AD to discover the k down-paths from the destination's core to the destination AD. Route joining combines these up and down paths to construct a working end-to-end route. The joining algorithm runs at the source endpoint AD, and is used to create an end-to-end "shortcut" path, if possible, between endpoint ADs. It takes as input the k up-paths of the source AD and the queried k down-paths of the destination, where each path is specified as a set of ingress/egress interfaces *and* peering interfaces at each transit AD as Section V describes.

The use of the peering links is to enable the source and destination to find not only joining points at a common ancestor AD, but also joining points at a common peering link. For example, in Figure 4, without considering the peering link between C and D, nodes F and G can only find route F, C, A, TC, B, D, G. When also comparing the peering nodes at the up-paths, F and G are able to find a shortcut path F, C, D, G.

The source endpoint needs to find a common ancestor provider or a common peering link to splice together the up-path of the source and the down-path of the destination AD. If the source and destination ADs are in distinct trust domains, then the route may need to traverse top-level TDs via the inter-TLTD protocol. Finding the common joining point can be accomplished via a number of methods. Since we anticipate that the paths will be reasonably short and k is also quite small (less than 10), a simple way is to hash the IDs of each provider and peering link of the destination AD down-paths into a hash table and look up the providers and peering links of the source AD. This takes time and space proportional to the total number of provider ADs and peering links named in both source and destination. For shortcuts crossing TD1 and TD2, the common join point is either an AD belonging to both of the TDs or a peering link with one end in TD1 and the other in TD2.

VIII. Forwarding

Once a source AD constructs an end-to-end route, it gathers the opaque fields for all transit ADs in the selected route, and embedds these opaque fields in its data packets to allow the transit ADs to verify the authenticity of the path and find the egress interface at each hop. The ingress and egress interfaces in the opaque fields instruct each transit AD on how to route the packets to the next hop, without requiring a forwarding table lookup. Each AD derives a MAC verification key from the timestamp TS embedded in the data packet, which is then used to check the MAC. If the MAC is correct we say that the link is not expired. The way opaque fields are gathered differs slightly depending on whether the path is a complete up/down path or a shortcut path:

1) The end-to-end route is the combination of a complete up-path and down-path. Here, the source AD embeds all the opaque fields constructed by the transit ADs in both the up-path of the source AD and the down-path of the destination AD. For example in Figure 4, AD E uses up-path $p_E = \{E, C, A, TC\}$ and down-path $p_G = \{TC, B, D, G\}$ to reach AD G, and needs to embed $O_{p_E}(C), O_{p_E}(A), O_{p_E}(TC), O_{p_G}(TC), O_{p_G}(B)$, and $O_{p_G}(D)$ into its data packets.

2) The end-to-end route is a shortcut. Once a common joining point X (either a common ancestor AD or a common peering link) is found between an up-path p_A of the source AD A and a down-path p_B of the destination B, the source only embeds the opaque fields of (i) the transit ADs in the shortcut, and (ii) the *immediate previous-hop* AD(s) of the joining point X. The opaque fields for (ii) are needed because according to Equation 2, the MAC in the opaque field $O_p(i)$ also includes the opaque field $O_p(i-1)$ from the immediate previous hop, thus requring $O_p(i-1)$ for MAC verification. To illustrate, consider ADs E and F in Figure 4 communicating through a shortcut $\{E, C, F\}$. The source AD E embeds $O_{p_E}(C)$, $O_{p_E}(A)$, $O_{p_F}(C)$, and $O_{p_F}(A)$ into its data packets, where p_E is E's up-path and p_F is F's down-path. Note that $O_{p_E}(A)$ and $O_{p_F}(A)$ may not be equal, because they may originate from path construction beacons with different timestamps TS. As another example in Figure 4, E and G communicate via the shortcut $\{E, C, D, G\}$. Then the source AD E embeds

$O_{p_E}(C), O_{p_E}(A), O_{p_G}(D), O_{p_G}(B)$, and the opaque field for the peering link $O_{C,D}(C)$ into its data packets.

Finally, the packet contains a pointer that advances at each AD to indicate which opaque field to use at that AD, and a pointer to indicate if the current AD is in the up-path or down-path to prevent routing loops and to make correct distinction between ingress and egress interfaces in the opaque fields. When the packet has reached the destination, the destination can reverse the paths to construct a symmetric return path, thus facilitating two-way communication. During the lifetime of the connection, the source endpoint can monitor path quality and switch to any of the other k^2 combinations of alternative path choices to improve latency, throughput, or drop rates.

IX. Security Analysis

Rather than designing our architecture with specific countermeasures against known attacks, we have designed it using sound principles of *isolation, control and explicit trust* such that security follows as an inherent property rather than as an add-on by separate sub-protocols. This section first presents SCION's intrinsic security against known attacks, and then discusses SCION's general attack resilience.

A. Defending against Known Attacks

We show that SCION can naturally eliminate or limit a wide range of severe control- and data-plane attacks.

Prefix (AID/EID) hijacking. SCION provides in-depth defense against prefix (AID/EID) hijacking. First, each endpoint AD or endhost uses AIP [17] as the self-certifying address. Second, the identities/addresses of the endpoint ADs or endhosts are scoped and isolated in different TDs and are signed by the corresponding TD Core. Hence, even if a malicious endpoint M in TD 1 claims the same AID or EID as an endpoint in TD 2, the claimed AID or EID is still scoped to TD 1 and will not collide with that in TD 2. Moreover, since M does not have the private key corresponding to the public key from which the identifier of AID is derived, M cannot sign any valid statements for the AD.

Routing path falsification. During path construction in SCION, each transit AD commits itself into the path construction beacon and signs both the locally announced link and the preceding path information in an onion fashion. Hence, a malicious AD cannot drop particular ADs in the preceding path to make the path shorter and more appealing, but can only drop the entire preceding path which would cause the endhosts to select other paths without the malicious router. Due to the use of onion signatures, a malicious router cannot extract and splice segments from different paths, nor can it modify the previous hops in the path.

Wormhole attacks. Two colluding ADs can announce a bogus link between each other, to create shorter paths for attracting traffic. SCION limits such attacks in two ways. First, if scoped path resolution (Section VI-B) is used for getting high-assurance paths (where the end-to-end paths must be constructed within the same TD), only two malicious ADs within the same TD can create a wormhole link between each other (since paths/links are restricted to particular TDs), and can only affect the traffic of their customers in that TD. Second, even if cross-domain paths are allowed (e.g., a shortcut with a cross-domain peering link), the endpoint ADs still know exactly who is on the communication path due to the signatures and certificates included in the down- and up-paths, thus retaining explicit trust.

Data-plane attacks. Malicious routers can drop or falsify packets at the data plane, including both control messages such as routing updates and data packets. SCION mitigates such data-plane attacks in several ways. First, SCION provides an endpoint AD with multiple path choices; thus an endpoint AD can efficiently avoid a path with detected poor performance. Second, each transit AD digitally signs itself into the path construction beacons and hence the endpoint AD knows exactly who is on the forwarding path (and is accountable for potential misbehavior), thus providing accountability for the ADs' forwarding behavior. Third, a malicious router M dropping path construction beacons can only render the path/link containing M unavailable to the endpoints (while the endpoints can select other paths excluding the malicious router), thus gaining itself no advantage.

Reflection DoS attacks. In a reflection attack, an attacker spoofs the return address of a primary target B to a secondary target A, so that A sends unwanted packets to target B. In systems that permit verifying address ownership, such as AIP, it is possible to add-on a protocol to authenticate return addresses (e.g., by ensuring that the packet is signed with the public key of the owner of the return address). This approach, however, adds complexity and overhead. For example, in return-address signing, the endpoint (or router) must verify a signature on every incoming data packet. SCION provides more inherent protection against reflection attacks, without needing cryptographic verification, because the way that packets are addressed is integrated with the path that it traverses. Consider a malicious AD M attempting to inject an attack packet to A with a spoofed return path to a legitimate target AD B. Since return paths are symmetric, M itself must be on the return path from A to B, so M has gained no advantage from the attack because it might as well have directly flooded the primary target B.

B. General Attack Resilience

Isolation: enabling attack localization. As discussed in Section II, the primary weakness of current inter-domain routing protocols is that the system is vulnerable to routing plane attacks by *any* adversary that is, or could make itself be, on any path between the source and the destination. In contrast, SCION provides strict isolation properties to localize attacks, failures, and misconfiguration. Since routing computation is isolated by trust domains, a malicious AD can only attack routes that have at least one endpoint inside its own trust domain. In addition, the victims of these routing attacks tend to be downstream endpoint ADs, who are the final approvers of the route computation in SCION (each endpoint must explicitly select k down-paths, sign the selection and upload

it to a route server). Because the target is actively involved in final route approval, attacking the route computation cannot be done stealthily: the target of the attack always has a chance to examine the forged route and always has to approve it explicitly. Compare this, for example, with the case of path computation in BGP, where a destination has no control over, and is often unaware of what routes to itself are eventually adopted by the rest of the Internet.

Control and explicit trust: providing resilience and flexibility. In SCION, an AD is able to determine or control the route that it uses to another AD, and is thus able to facilitate a well-quantified level of trustworthiness or reliability for its network service provision. Specifically, both endpoint ADs in end-to-end communication can select a set of k well-defined paths to the TD Core. This choice is made explicitly, with knowledge of the exact identities supplying the path and the full authentication information of each path is provided. A source AD gets both these sets to choose from, yielding potentially up to k^2 end to end AD-level paths. A destination AD can in fact select different sets of k paths to serve to different source endpoints; for example, the destination AD can approve a separate set of high-assurance paths for trusted entities; this can be provided even if the TD route servers are untrusted by encrypting the route record using a group key. Furthermore, these route sets are all separated by trust domain, with each domain maintaining a different set of k paths. An AD that is in more than one TD can thus not only switch paths but also change the routing context to a different TD.

Scalability: route freshness. Current inter-domain routing protocols are based on path-vector instead of link-state routing, partially because in link-state routing each node must periodically generate routing updates which are propagated throughout the entire network. However, in path vector, routing updates are generated "on-demand" only when route changes happen to achieve scalability. Attempting to secure these incremental updates is problematic. An attacker could re-order or re-inject route update messages causing invalid and inconsistent routes to propagate in the network. Path-vector based route announcements cannot have short timeouts, since a path-vector update requires a destination AD to push its announcement to the entire network, and the protocol is not scalable if every AD is performing this broadcast at a high rate like link-state routing, since this would cause $O(n^2)$ communication overhead per update where n is the number of network nodes (since it involves all-to-all communication).

In SCION on the other hand, all route updates (the path construction beacons) originate directly from the destination AD via the route servers of the TD. There is no global distributed consistency issue since the route servers are centrally administered. Since all route discovery occurs in an upstream-to-downstream direction, frequent updates can be done highly scalably ($O(k \cdot n)$ communication per update) in a coordinated fashion. Hence, published routes can have very short timeouts and be updated frequently and securely.

TABLE II
RESULTS OF TRUST DOMAIN FORMATION.

	Trust Domain	# of ADs	# of TD Core ADs
TD1	Africa (AfriNIC)	613	39
TD2	America (ARIN)	21619	38
TD3	APAC (APNIC)	6039	29
TD4	LATAM (LACNIC)	1912	60
TD5	RIPE NCC	19569	34

X. EVALUATION

Due to the infeasibility of evaluating a completely new architecture on the current Internet, we have constructed an AD topology based on real-world datasets to evaluate the effectiveness of SCION. Specifically, we simulate SCION on a measured Internet AD topology annotated with the business relationships from a CAIDA dataset[1] to evaluate the routing efficiency, security, and expressiveness.

A. Evaluation Methodology

Trust domains. Given a measured AD topology from CAIDA, we group the ADs into several trust domains and assign some of the ADs as the TD Core ADs in order to simulate SCION Trust-domain-based routing. In this proof-of-concept evaluation, we virtually divide the Internet into five local and non-overlapping trust domains, and each of these five local TDs associates with one Regional Internet Registry (RIR), the regional organization allocating AD numbers. In other words, ADs registered to the same RIR belong to the same trust domain in our experiment. Such a division reflects the geographical and administrative relationships to some extent. Table II summarizes the size of each of these trust domains. The TD Core ADs are defined as TD Core ADs that have no providers themselves in their respective trust domain.

BGP Routing. When simulating BGP (and S-BGP) routing, we assume that in benign cases ADs respect and make routing decisions based on the business relationships with their neighbors, and then use path length as the tie-breaking factor. We also assume that the TD Core ADs form a clique, and thus the length of any inter TD Core routing path is 1. This is accomplished by adding a peering link (if it does not yet exist) between every pair of TD Core ADs.

Finding k up-paths. In practice, each AD running SCION can have different policies in determining which k paths to export, the value of k, and the algorithm for finding the (presumably disjoint) k paths. However, the optimization of the export policies, the k value, and the finding-disjoint-path algorithm are outside the scope of this paper. Instead, for the purpose of simulation, we implement a simple k-path discovery algorithm that takes a source AD, the complete AD-level topology, and a Trust Domain as inputs, and yields a set of k *maximally edge disjoint paths* to the TD Core ADs in the specified Trust Domain. Specifically, our k-up-path

[1]CAIDA. http://as-rank.caida.org/data/

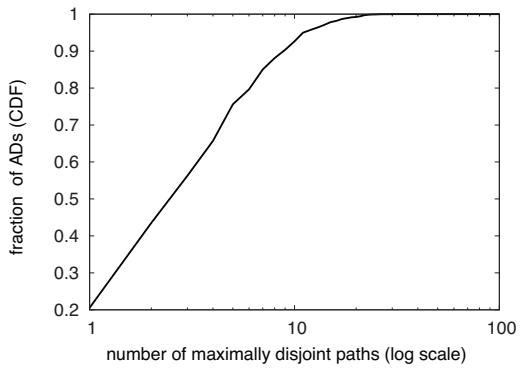


Fig. 5. Measurement results of AD-level end-to-end path diversity.

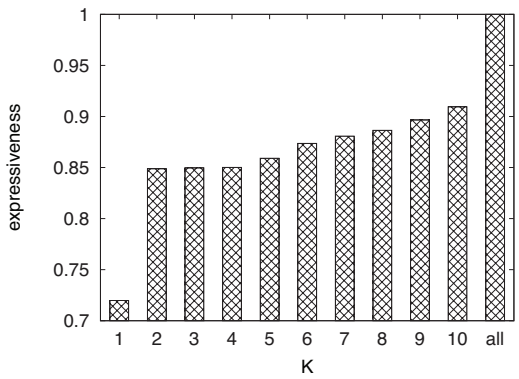


Fig. 6. Measurement results of SCION expressiveness.

TABLE III
SCION PATH STRETCH COMPARED TO BGP AND PATH LENGTH FOR ROUTING IN A TD WITH SHORTCUTS ENABLED.

K	1	2	3	4	5
SCION path stretch	1.067	1.035	1.035	1.029	1.025
SCION path length	3.407	3.385	3.383	3.322	3.286

discovery algorithm selects the disjoint paths using an iterative, greedy algorithm. At step i, the greedy algorithm 1) finds the current shortest path as the i^{th} maximally disjoint path, and 2) increases the weight of all the edges on the i^{th} path such that these edges become less preferred in the next iteration. Fig. 5 shows the distribution of the number of available paths. More than 90% of ADs have fewer than 10 available paths in their Trust Domain, which indicates that the Internet has a shallow AD-level topology.

B. SCION Shortcut Efficacy: Route Stretch

SCION uses shortcuts to reduce the route length; when the source and destination endpoints have a common provider or a common peering link, they can communicate directly without traversing TD Core. In this experiment, we evaluate the effectiveness of this shortcut mechanism in the intra-TD routing using the *end-to-end route stretch*: the ratio of the length of a SCION route to the length of a BGP route for a given pair of source and destination endpoints. We focus on routing within a TD which represents the common case with fully protected route computation. Our simulation takes 1000 random pairs of source and destination ADs in the same TD. For each pair, we measure the length of its BGP route and SCION route. Table III summarizes the average length of SCION paths and its stretch. The result demonstrates that SCION paths only add a small amount of overhead in terms of path length compared to BGP. Also, SCION paths become shorter as k increases because endpoints are more likely to find a common AD or link for shortcut construction when they have more up-paths.

C. Route Construction Efficacy: Expressiveness

In SCION, route update is efficient and scalable because only the TD Cores need to announce their reachability information, in contrast to path vector where every AD floods its update to potentially every other AD in the Internet. However, we are interested in knowing whether our gain in scalability comes at the cost of path expressiveness compared to path vector.

In this experiment, we show that given SCION's well-isolated TD infrastructure, a route discovered by BGP-style (i.e., flooding) route updates is likely to be found through SCION's scalable route update propagation as well. Specifically we define *trust-scoped expressiveness* as the fraction of source and destination pairs whose "trusted" BGP path is discoverable in SCION. "Trusted" BGP paths are intra-domain paths or inter-domain paths that pass through the TD Core.

The evaluation proceeds as follows: we randomly select 1000 pairs of source and destination ADs in the same TD. For every pair, we compute the trusted BGP path between the pair as well as the SCION up-paths of the source and the destination ADs, and check whether the BGP path between the pair is contained (i.e., in the union of the source's up-paths and the destination's down-paths). The ratio of contained paths represents the expressiveness. We evaluate the expressiveness as a function of k based on our simple k-path selection algorithm to demonstrate the practicability of SCION and the trend as k increases. Fig. 6 summarizes the results of SCION's expressiveness experiments, from which we can see that with only a $k = 5$, SCION can already capture more than 85% of BGP paths.

D. Security

Section IX discussed attacks that are naturally infeasible in SCION. In this section, we quantitatively investigate how severe such attacks are in networks lacking our well-defined properties. As an example, we consider the impact of traffic attraction attacks where the attacker attempts to attract routes by announcing a non-existing shortcut (or "wormhole"). Clearly, with SCION's strong isolation property, it is infeasible for an outsider to pull traffic out of a TD, whereas in a network without trust-based isolation, a wormhole residing in any corner of the Internet can possibly attract a significant portion of traffic and eavesdrop on unencrypted communication.

In the simulation, a group of colluding ADs (except the TD Core ADs that we assume trusted) announce a link with minimum cost between each other to attract traffic. We consider

TABLE IV
PERCENTAGE OF TRAFFIC BETWEEN TWO TDS DIRECTED TO MALICIOUS ADS IN AN UNTRUSTED TD (TD_m), SELECTED RANDOMLY FROM ALL TDS EXCEPT THE SOURCE AND DESTINATION TDS. ASSUMING THE COMMUNICATION IS SYMMETRIC. SUCH INCIDENTS CAN BE COMPLETELY PREVENTED IN SCION BECAUSE OF SCION'S STRONG ISOLATION PROPERTY.

(a) Attraction attack scenario 1. The communication between the source and destination TDs is attacked by all ADs in an untrusted TD.

		[Destination TD]				
		TD1	TD2	TD3	TD4	TD5
[Source TD]	TD1	34.3%	18%	45.7%	37%	18.6%
	TD2	-	1.2%	9.3%	4%	1%
	TD3	-	-	26.2%	38%	23.3%
	TD4	-	-	-	17.5%	18.3%
	TD5	-	-	-	-	7.5%

(b) Attraction attack scenario 2. The communication between the source and destination TDs is attacked by the ten most influential ADs in an untrusted TD.

		[Destination TD]				
		TD1	TD2	TD3	TD4	TD5
[Source TD]	TD1	18.2%	6.7%	27.1%	26%	15%
	TD2	-	0.2%	4.3%	2.4%	0.6%
	TD3	-	-	13.5%	23.9%	20.4%
	TD4	-	-	-	11.5%	17.1%
	TD5	-	-	-	-	6.8%

such attacks by the most influential ADs in an untrusted TD, TD_m. The influence score of an AD in TD_m is evaluated by the number of ADs seeing this AD on the shortest path to TD_m. Our simulation repeats 1000 random selections of source and destination ADs. In each run, TD_m is selected randomly from all TDs except the source and destination TDs. We measure the fraction of traffic between these two TDs being redirected to TD_m.

Table IV summarizes the impact of malicious ADs on the path-vector-based routing protocols without isolation (BGP/S-BGP for example). Each of the data values is an average over 1000 runs of simulation with randomly selected pairs of source and destination. Without strong isolation, the Internet is fragile: an attacker can control a significant fraction of traffic with only ten compromised ADs. In contrast, by leveraging the isolation principle, SCION can intrinsically mitigate these attacks.

XI. DISCUSSION

Accountability. Deterrence through accountability can be a powerful mechanism for security. By grouping networks and hosts into trust domains with a common legal or contractual framework, SCION helps ensure not just that malicious actors can be identified, but that they can be held accountable in a meaningful way. Specifically, SCION provides accountability for path construction information, as the digital signatures and certificates provide proof of origin for the path information. A weaker notion of accountability is also provided by opaque fields used for packet forwarding: the path that a packet traversed so far leads back to the sender, and thus, a MAC field that does not match implies that a malicious forwarder or sender is among the preceding entities on that path. Unfortunately, the lack of digital signatures in the forwarding path prevents more specific attribution.

Error message propagation. Erroneous paths, i.e., a link failure, can be handled actively or passively. In a passive approach, an AD would cease to announce paths containing that link and let paths containing the link naturally time out. Unfortunately, packets forwarded across that link would be dropped and senders would need to monitor such failures and resort to a different path – although without any explicit notification senders would not know the reason for the packet loss nor the fault location. We prefer a more active approach, where link failures would be actively announced to all neighbors who received path construction beacons containing those links. End domains can thus stop using up-paths containing those faulty links, and remove down-paths containing these links on the path server.

Number of trust domains. We anticipate a small number of top-level trust domains for reasons of efficiency. Running a trust domain is expensive because of the maintenance of address and path resolution servers, as well as the key management and certification authority requirements. Given economies of scale, the larger the TD the better the fixed costs become amortized. Moreover, larger TDs provide more opportunities for shortcuts, i.e., more efficient paths, which again favors large TDs.

Incentives for adoption. SCION offers many incentives for adoption. For ISPs, network operations are simplified and costs are reduced: (i) explicit forwarding paths enable fine-grained route control without changing router configurations; (ii) prevention of control-plane attacks also provides resilience against router misconfigurations, which are a frequent reason for network outages [18]; (iii) ISPs can isolate control-plane messages from other ISPs for which no enforceable recourse is available, and can (iv) validate forwarding information. (v) Finally, SCION may enable simpler routers, as complex route table lookups are not necessary any more.

For senders and receivers, SCION offers path control and explicit trust, because end-points can make differentiated trust decisions based on the different forwarding paths that are available.

Granularity of path choice for senders and receivers. SCION offers path choice at an intermediate granularity, where BGP is at a course granularity offering no path choice to senders and receivers, and proposals such as Pathlets [4] offer very fine-grained path choice. An advantage of intermediate granularity is that distribution of path information is limited to entities who really want to use these paths, thus resulting in a low overhead of path distribution and consequently better scalability. From a security perspective, fine path granularity is dangerous, because attackers can potentially create path loops that can focus and amplify traffic onto a specific network area. Finally, verifying the adherence of paths to the policy of ADs is more difficult for finer granularity proposals. Given that SCION offers on the order of k^2 path choices for point-to-point links, it appears that path choice is plentiful for most

applications without introducing potential security and policy challenges.

Incremental Deployment. Although SCION differs significantly from the way the current Internet functions, we can envision a deployment path that requires relatively modest changes. First, we observe that the current ISP topologies are consistent with the TDs in SCION, as top-tier ISPs are the providers for smaller ISPs in a geographic area, and the top-tier ISP connect to other top-tier ISPs in other areas. As a result, we anticipate that traffic flows in SCION will closely resemble current traffic flows. Furthermore, current ISPs make use of MPLS to forward traffic within their networks. Requiring only changing some edge routers to SCION-enabled routers, these edge routers can perform all the SCION-related processing and utilize MPLS to forward traffic to the desired egress point (note that the opaque field already contains the ingress and egress points, and thus, the required processing is quite minimal). SCION-enabled edge routers in different autonomous domains do need to be connected to each other, for which we can use IP-tunnels in case they are not directly adjacent. To route to destinations identified by an EID within an AD, either the intra-domain routing protocol will support a flat name space, or SCION-enabled end hosts can open an IP tunnel to a SCION-enabled edge router.

Hence, SCION possesses a natural deployment path when overlaid on the current Internet, requiring only those entities gaining immediate benefit to incur costs.

Data-Plane DoS Resilience. While this paper focuses on control-plane issues, we briefly identify architectural features of SCION that can enable powerful DoS defenses: (i) the separation of path construction and forwarding protects existing paths from control-plane disruptions, as forwarding paths can continue to be used even while the control plane is dysfunctional; (ii) opaque paths can be used to encode stateless capabilities [19] in a seamless manner; (iii) forwarding paths in packets provide a return path to the source and thus prevent source address spoofing naturally, enabling a "shut off" message to reach undesired senders; (iv) while multi-paths provide attackers more opportunities, they also give receivers additional options, such as keeping some disjoint paths secret which can be used during an emergency when the announced paths are under attack; (v) announced down-paths can be routed through a filtering cluster to remove malicious DDoS traffic, without requiring configuration changes on forwarding routers.

AD Key Management. SCION requires signatures on route construction beacons, which requires access to a private key to compute the signature. A problem is that the disclosure of the private key would have severe consequences for the AD. Thus, we propose a hierarchy of keys as proposed by DNSSEC [20], where the domain's long-term private key resides on an off-line system, but certifies shorter-term keys which reside on routers that need to sign path construction beacons.

Expressiveness of the opaque field. The opaque field is constructed by each transit AD to enable efficient forwarding. In addition to including ingress/egress interfaces to dictate a forwarding path, the opaque field can also include other information for the corresponding AD to implement flexible routing policies and traffic engineering. For example, an AD can encode unidirectional local links in the opaque field, so that endpoint ADs can only use a certain local link in one direction. To support different expiration times within an AD, expiration markers can be added as well.

XII. Related Work

While no existing solution *simultaneously* provides routing security, control, isolation, and explicit trust as SCION does, prior work has attempted to address individual routing problems as summarized below. SCION builds upon numerous ideas from these efforts.

Routing security. Goldberg et al. analyze the weaknesses of BGP and S-BGP [2], quantifying their efficacy in defending against traffic attraction attacks. Indeed, existing secure routing protocols such as soBGP [21], psBGP [22], SPV [23], and PGBGP [24] only address the security of route announcement *semantics*, which, at best, only guarantees the paths are *topologically* valid but fails to to ensure the *logical* trustworthiness and *contractual* legitimacy of the routes.

Routing control. A number of proposals aim to give a *source* node more control over which paths to use for end-to-end communications via source routing [25] and multi-path routing [4], [6], [7], [10], [26], [27]. However, in these protocols, the destinations, which are also the primary stakeholders for end-to-end communications, are still incapable of implementing inbound traffic control. In NIRA [15], each endpoint also discovers and uploads paths for reaching the "core" of the Internet. A source endpoint can thus query back the paths uploaded by the destination endpoint for reaching the Internet core, and construct an end-to-end communication path. NIRA can thus provide route control for both source and destination endpoints. SCION uses a similar style of path construction, but adds the cryptographic and architectural means to defend against attacks, provide isolation, and ensure that trust is explicit.

Routing isolation. The previously proposed HLP [28] divides the Internet into a set of isolated regions, within each of which link-state routing is used. These regions are further interconnected by path-vector routing, and the routing failures in one link-state region are invisible to other link-state regions, thus providing isolation of routing failures and churns. However, HLP does not employ cryptographic mechanisms to defend against attacks, and endpoints have little control over which communications paths to use.

Explicit trust and minimal TCB. The high-level philosophy of SCION mirrors that of trustworthy computing: we want to explicitly delineate which components are trusted and which are not for a communications system, and the untrusted components cannot tamper with the computation within the trusted components. The trusted components constitute the Trusted Computing Base (TCB) for the system, and it has been recognized that a small TCB can enable better security [29]. In SCION, only the TD Cores are trusted and thus constitute a small TCB.

Next-generation Internet architectures. While SCION aims to provide intrinsic security for the future network architecture, there are a number of other proposals attempting to achieve other goals orthogonal to ours. For example, AIP [17] intends to provide accountable and secure identifiers/addresses for network hosts. As another example, rule-based forwarding (RBF) [30] introduces a new architectural concept called *packet rules* (each rule is a simple if-then-else statement). In RBF, instead of sending packets to a destination (IP) address, end-hosts send packets using the destinations rule.

XIII. Conclusion and Future Work

Splitting up networks into separate trust domains (TDs) provides surprisingly useful properties: it enables strong isolation from external events and supports meaningful accountability and enforceability within a TD because every TD is governed by a uniform legal framework. TD separation also resolves the problem of a single root of trust, which has plagued previous proposals. By segregating mutually distrustful entities, each TD can choose a coherent trust anchor (e.g., few tier-1 ISPs) that everyone in the TD agrees to trust. Consequently, an entity only has to trust a small subset of the network thus achieving a small TCB.

This TD infrastructure, SCION, enables a design of an AD-level routing protocol that supports scalable route update propagation without flooding per-destination updates and mutually controllable path selection at an appropriate granularity to trade off route control with attack power. Also as SCION makes trust explicit through isolation and cryptographic primitives (i.e., signatures and MACs), entities know who is accountable for incorrect path construction and message propagation. Moreover, SCION is designed with forward extensibility and backward compatibility in mind. Through the use of opaque fields in the packet header, SCION is flexible to unforeseen extensions and is agnostic to the underlying routing protocols within ADs. Also, incremental deployment is possible because SCION is compatible with the current Internet topology and ISPs' business relationships.

Based on the well-defined foundation of the SCION Internet architecture and routing, the next step is to work out the details of the data plane mechanisms and DoS defense, secure and scalable name lookup, revocation and update of keys in the AD and TD Core, and privacy issues.

Acknowledgments

We gratefully thank John Byers, Virgil Gligor, Marco Gruteser, Srini Seshan, Peter Steenkiste, and Hui Zhang for constructive discussions and insightful suggestions, and the anonymous reviewers for their valuable feedback.

References

[1] "Insecure routing redirects youtube to pakistan," February 2008, http://arstechnica.com/old/content/2008/02/insecure-routing-redirects-youtube-to-pakistan.ars.

[2] S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure border gateway protocol (S-BGP) — real world performance and deployment issues," in *Symposium on Network and Distributed Systems Security (NDSS)*, Feb. 2000.

[3] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford, "How secure are secure interdomain routing protocols," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 87–98, 2010.

[4] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica, "Pathlet routing," in *In Proc. SIGCOMM Workshop on Hot Topics in Networking*, 2008.

[5] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala, "Path splicing," in *ACM SIGCOMM*, 2008.

[6] W. Xu and J. Rexford, "MIRO: Multi-path Interdomain Routing," in *ACM SIGCOMM*, 2006.

[7] N. Kushman, S. Kandula, D. Katabi, and B. M. Maggs, "R-BGP: Staying Connected In a Connected World," in *USENIX NSDI*, 2007.

[8] X. Zhang, A. Perrig, and H. Zhang, "Centaur: A hybrid approach for reliable policy-based routing," in *Proceedings of the International Conference on Distributed Computing Systems (ICDCS)*, Jun. 2009.

[9] X. Yang and D. Wetherall, "Source Selectable Path Diversity via Routing Deflections," in *ACM SIGCOMM*, 2006.

[10] D. Wendlandt, I. Avramopoulos, D. Andersen, and J. Rexford, "Don't secure routing protocols, secure data delivery," in *ACM Hotnets*, 2006.

[11] X. Zhang and A. Perrig, "Correlation-resilient path selection in multi-path routing," in *IEEE Globecom*, 2010.

[12] M. Caesar and J. Rexford, "BGP routing policies in ISP networks," *IEEE Network Magazine*, vol. Special issue on interdomain Routing, Dec 2005.

[13] L. Gao and J. Rexford, "Stable internet routing without global coordination," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 681–692, 2001.

[14] R. Mahajan, D. Wetherall, and T. E. Anderson, "Understanding BGP misconfiguration," in *ACM SIGCOMM*, 2002.

[15] X. Yang, D. Clark, and A. W. Berger, "NIRA: a new inter-domain routing architecture," *IEEE/ACM Trans. Netw.*, 2007.

[16] P. F. Tsuchiya, "The landmark hierarchy: a new hierarchy for routing in very large networks," in *Proceedings of ACM SIGCOMM*, 1988.

[17] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, "Accountable Internet Protocol (AIP)," in *Proceedings of ACM SIGCOMM*, Aug. 2008.

[18] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," in *ACM SIGCOMM*, Aug. 2002.

[19] A. Yaar, A. Perrig, and D. Song, "SIFF: A stateless Internet flow filter to mitigate DDoS flooding attacks," in *IEEE Symposium on Security and Privacy*, 2004.

[20] O. Kolkman and R. Gieben, "DNSSEC Operational Practices," 2006.

[21] R. White, "Securing BGP through secure origin BGP," Cisco Internet Protocol Journal, Tech. Rep., Sep. 2003.

[22] T. Wan, E. Kranakis, and P. van Oorschot, "Pretty secure BGP (psBGP)," in *Proceedings of Symposium on Network and Distributed System Security*, 2005.

[23] Y.-C. Hu, A. Perrig, and M. Sirbu, "SPV: Secure path vector routing for securing BGP," in *Proceedings of ACM SIGCOMM*, Sep. 2004.

[24] J. Karlin, S. Forrest, and J. Rexford, "Pretty good BGP: Improving BGP by cautiously adopting routes," in *Proceedings of IEEE International Conference on Network Protocols*, 2006.

[25] K. Argyraki and D. R. Cheriton, "Loose source routing as a mechanism for traffic policies," in *ACM SIGCOMM Future Directions in Network Architecture*, 2004.

[26] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy, "Routing amid colluding attackers," in *IEEE ICNP*, 2007.

[27] X. Yang and D. Wetherall, "Source selectable path diversity via routing deflections," in *ACM SIGCOMM*, 2006.

[28] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica, "HLP: a next generation inter-domain routing protocol," in *ACM SIGCOMM*, 2005.

[29] J. M. McCune, B. Parno, A. Perrig, M. K. Reiter, and H. Isozaki, "Flicker: An execution infrastructure for TCB minimization," in *Proceedings of the ACM European Conference in Computer Systems (EuroSys)*, Apr. 2008.

[30] L. Popa, I. Stoica, and S. Ratnasamy, "Rule-based forwarding (RBF): improving the internets flexibility and security," in *ACM HotNets*, 2009.