Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 50 (2006) 2953–2980
Computer Networks
www.elsevier.com/locate/comnet

Origin authentication in interdomain routing

Patrick McDaniel [a], William Aiello [b], Kevin Butler [a,*], John Ioannidis [c]

[a] Computer Science and Engineering, Pennsylvania State University, 344 IST Building, University Park, PA 16802, United States
[b] Department of Computer Science, University of British Columbia, 201 Main Mall, Vancouver, Canada, BC V6T 1Z4
[c] Center for Computational Learning Systems, Columbia University, 475 Riverside Ave, New York, NY 10115, United States

Received 11 April 2005; received in revised form 27 July 2005; accepted 19 November 2005
Available online 28 December 2005

Responsible Editor: L.G. Xue

Abstract

Attacks against Internet routing are increasing in number and severity. Contributing greatly to these attacks is the absence of *origin authentication*; there is no way to validate claims of address ownership or location. The lack of such services not only enables attacks by malicious entities, but also indirectly allows seemingly inconsequential misconfigurations to disrupt large portions of the Internet. This paper considers the semantics, design, and costs of origin authentication in interdomain routing. We formalize the semantics of address delegation and use on the Internet, and develop and characterize original, broad classes of origin authentication proof systems. We estimate the *address delegation graph* representing the current use of IPv4 address space using available routing data. This effort reveals that current address delegation is dense and relatively static: as few as 16 entities perform 80% of the delegation on the Internet. We conclude by evaluating the proposed services via trace-based simulation, which demonstrates that the enhanced proof systems can significantly reduce resource costs associated with origin authentication.

Keywords: Routing; Security; Address management; BGP; Delegation; Authentication

1. Introduction

Routing within the Internet dictates the path that IP packets take to get from their source to their destination. In its most general form, this path, called the route, is a sequence of routers and the links between them. To compute such paths, routers use a *routing protocol* to exchange reachability data, and perform computations on these data to compute the desired routes. Computing the correct route is a complicated task because of the sheer scale of the problem; several hundred thousand routers have to perform a distributed computation that must produce compatible results. The issue of scale is somewhat mitigated by considering the Internet as consisting of many *routing domains*; routing inside a domain is determined by an *intradomain* routing protocol, while routing between domains is governed by an *interdomain* routing protocol.

* Corresponding author. Tel.: +1 814 865 6245.
E-mail addresses: mcdaniel@cse.psu.edu (P. McDaniel), aiello@cs.ubc.ca (W. Aiello), butler@cse.psu.edu (K. Butler), ji@cs.columbia.edu (J. Ioannidis).

1389-1286/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2005.11.007

Intradomain and interdomain routing decisions are largely made independently.

The Border Gateway Protocol [49,55] is the interdomain routing protocol used on the Internet. BGP routing domains called *Autonomous Systems* (ASes) announce IP address ranges, called *prefixes*, to their neighboring ASes. Each AS also announces the prefixes that it learns from each of its neighbors to its other neighbors.

The design of BGP reflects its egalitarian origins: ASes are trusted to behave per specification and to perform due diligence in providing timely and accurate routing information. In other words, BGP does not currently provide security. The need for security in interdomain routing has been widely acknowledged and evaluated [54,29,40,15], and interim and long-term solutions are seeking broad adoption [28,15,11]. Implemented by any comprehensive routing security solution, an *origin authentication*[1] (OA) service validates the delegation of address space between address authorities (e.g., IANA [24]), organizations, and advertising ASes. Origin authentication is fundamentally grounded in ownership: the address may be originated by an AS only if the owner has granted it the right to do so.

The lack of authenticated origin information is increasingly viewed as a critical vulnerability of the Internet infrastructure [16]. In one widely documented example, AS7007 announced it was the origin for large portions of the IPv4 address space. As a result, a huge part of the address space was incorrectly routed to that AS, which was not equipped to process the amount of traffic that was consequently generated. This led to widespread outages [39]. Similarly, Zhao et al. found that there are numerous situations where multiple ASes claim to be the origin of a single prefix (called a MOAS conflict), almost all of them anomalous [61]. The authors found that *prefix hijacking* due to apparent misconfiguration was a frequent cause of MOAS conflicts. Other outages were similarly enabled by incorrect origin and routing information [33].

This paper considers the semantics, design, and application of origin authentication services. We begin by formalizing the semantics of address delegation. An *address delegation graph* represents the delegation of IPv4 addresses from address authorities to organizations, and ultimately to ASes. We show that the semantics of address delegation mandates that any path (i.e., delegation chain) in this directed graph adheres to the following: (a) the origin of the path is IANA, (b) the path is acyclic, and (c) the last node in the path is an AS. In the origin authentication systems considered in this paper, entities delegate address space by generating and distributing proofs reflecting edges in the graph. To simplify, an OA proof is a signed statement asserting that: (a) an organization has been delegated authority (by IANA or some organization) over a specified address range, (b) an AS has been granted the right to be the origin of that address range, or (c) the address range cannot be used (reserved). Verifiers collect and validate proofs corresponding to the delegation chains. We apply a range of novel cryptographic constructions that we have devised to the problem of proof construction and consider the complexities of their application in real environments.

While identifying constructions that meet the semantic requirements of origin authentication is a useful and necessary endeavor, one must also evaluate their feasibility. However, any evaluation of this sort must be informed by an understanding of the current use of the IP address space. We develop an approximate address delegation graph for the Internet from public data. One of the key results of this investigation shows that the delegation of IP address space is exceptionally dense: 80% of delegation is performed by 16 entities in our approximate graph, and 90% by 122. Moreover, these delegations evolve slowly. Such results are encouraging: proof systems are most effective where the bulk of delegation is both static and dense.

It has been argued that in-band origin authentication is inherently infeasible. We compare the costs of in-band and out-of-band mechanisms via trace-based simulation. Our *OAsim* simulator models a BGP speaker implementing several OA service designs using the approximate address delegation graph and collected BGP update stream data. Our simulations uncover two central results. First, the efficiencies afforded by our origin authentication designs make in-band verification possible. For example, an in-band *authenticated delegation tree* uses as little as one-tenth the computational resources of current solutions. Second, we found that proof systems consolidating proofs by delegator can significantly reduce resource costs.

[1] We use the term *origin* to refer to the AS in which a set of addresses resides. This is not to be confused with the *origin attribute* of BGP, which specifies the source of routing information (e.g., eBGP/iBGP).

This work is not intended as a replacement for comprehensive interdomain routing security infrastructures. We do not specifically address path or attribute validation. Hence, this work addresses only one aspect of the larger interdomain routing security problem: the creation and validation of proofs of ownership and origination. The designs and results described throughout are applicable to any such interdomain routing security service (e.g., S-BGP [29], IRV [15], soBGP [11,10]).

The remainder of this paper explores the design and practical use of origin authentication services. We begin in the following section by describing how address space is currently delegated.

2. Address management

The IPv4 address space is governed by IANA[2] [24]. IANA *delegates* parts of the global address space to organizations representing commercial, public, or other interests [59]. Each organization is free to further delegate some or all of the received address space to any organization it desires, but is prohibited from delegating the same address to more than one organization.

BGP is not aware of the existence of organizations. Autonomous systems advertise the set of prefixes that they originate (i.e., the addresses within their administrative domain). While many organizations maintain their own AS, many do not, and still others (typically connectivity providers) maintain more than one. Each organization may *assign* its address space to the AS in which the addresses reside. Hence, assignment is the process where an organization gives an AS the right to originate a set of addresses. Fig. 1 illustrates several common ways that address space is delegated to organizations and assigned to ASes.

In the early days of IP, IANA directly delegated address space to organizations. For example, as shown in the figure, AT&T received `12.0.0.0/8` directly from IANA in the 1980s. As the popularity of IP grew, it was determined that having a single body governing all delegation was administratively difficult. Hence, registries like ARIN [5] were introduced to delegate address space received from

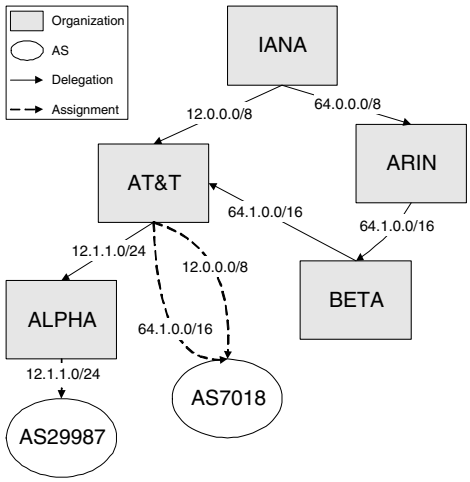

Fig. 1. IPv4 address management—All ownership of IPv4 address is *delegated* by IANA to organizations that may delegate further. Addresses are *assigned* to an AS for advertisement via BGP.

IANA. Organizations, such as BETA in the figure, currently request and receive address space from the registries (i.e., `64.1.0.0/16`). Assume that BETA is a customer of the provider AT&T, and that BETA's network is serviced by AT&T's AS. BETA delegates their address space to AT&T for the explicit purpose of providing service. The practical limitation of this "provider" delegation classification is that AT&T is barred from delegating the address further.

In practice, organizations are often delegated address space by their provider networks. For example, consider an organization DELTA (not shown) that is a customer of AT&T. Assume that DELTA is given its address space by AT&T and wishes to be part of AT&T's AS. In this case, there is no need for delegation because DELTA's address space is totally encompassed by AT&T (both in the logical and physical sense). Now consider another organization ALPHA that is also a customer of AT&T but wishes to run its own AS. ALPHA may wish to be its own AS to allow multi-homing or simply to retain control over the interdomain routing policy associated with its network. AT&T delegates parts of its address space to ALPHA (e.g., `12.1.1.0/24`) so ALPHA's AS can independently advertise the addresses (e.g., as may be desirable for multi-homing).

Assignment associates the addresses delegated to an organization with the ASes owned by it. These addresses are configured into routers that

[2] The IANA function is currently contracted to the Internet Corporation for Assigned Names and Numbers (ICANN), which some cite as the relevant authority. Throughout, we refer to IANA interchangeably to refer to both the ICANN organization and the IANA address authority function.

subsequently advertise them via BGP. From the figure, AT&T assigns the addresses it is delegated to the ASes under its control (e.g., AS7018 is assigned `12.0.0.0/8` and `64.1.0.0/16`), as does ALPHA (AS2997 is assigned `12.1.1.0/24`).

AT&T retains control of (originates) `12.0.0.0/8` by assigning the prefix to AS7018. This assignment is seemingly ambiguous: because `12.0.0.0/8` is a superset of `12.1.1.0/24`, they both assert control over the same addresses. This is resolved in BGP by the *longest prefix matching* rule: the longest prefix delegation/assignment (in terms of mask size) supersedes all shorter prefixes. Hence, AT&Ts delegation and ALPHAs subsequent assignment of `12.1.1.0/24` is always taken as authoritative over the assignment of `12.0.0.0/8`.

Delegation and assignment on the Internet is currently an administrative process. There is no structure for validating claims of address ownership and assignment. This paper addresses this need by attempting to both clarify the semantics of these assertions and define efficient constructions for their authentication.

A prerequisite of this work is a parallel management structure for the secure management of organizations and AS identifiers and associated cryptographic material. Seo et al. have considered such infrastructures in depth [53]. We assume an infrastructure for registering address authorities and organizations, as well as for the management of certificates assigned to these entities. Furthermore, authentication of speaker identity, and more generally of any aspect of the AS topology or path information, is explicitly outside the scope of this work.

3. Related work

Early works in interdomain routing security characterized the relevant threats and countermeasures [54,9,48], while recent works have provided summaries of fundamental problems within BGP [40,45,8,47]. The vulnerabilities of BGP can be classified in the following manner:

1. Messages do not have guaranteed integrity, freshness, or authenticity, leaving them vulnerable to attacks that can be carried out between two BGP hosts, such as man-in-the middle or session termination attacks.
2. Paths are not authenticated, leaving BGP messages susceptible to a malicious AS seeking to spread misinformation of routing data through the Internet, or divert traffic to a conspiring malicious AS.
3. There is no way to validate an AS's authority to advertise a prefix, leading to the potential for *prefix hijacking*, where a malicious AS advertises a prefix originated from another AS as its own, thus misdirecting traffic. This can also lead to *black holes*, where traffic intended for the legitimate AS is instead forwarded to the malicious AS, which can analyze or arbitrarily drop the packets.

The first item is addressed by IPsec [27]. The second item is considered in [20,2,19,44,57]. Additionally, these items are addressed by the three comprehensive approaches to BGP security, discussed in greater detail below. This paper focuses solely on the last item, the lack of authenticated address usage. Origin authentication traces the delegation of address space between authorities (e.g., IANA), organizations (e.g., IBM), and ASes. Seo et al. uncovered the hidden complexity in the delegation of not only IP addresses, but of other aspects of the interdomain routing (e.g., AS numbers) [53]. The natural and almost universally accepted method for tracing delegation in these large, complex networks is through signed assertions. In practice, the scale of the Internet mandates that these assertions be supported by a certification infrastructure.

A leading candidate for securing Internet routing, the comprehensive S-BGP extension to BGP addresses a wide range of threats [29,28,26]. Origin authentication is supported in S-BGP by an address allocation public key infrastructure (PKI). Authorities in the S-BGP PKI issue certificates binding prefixes to organizations (e.g., IANA delegates part of an address space to ARIN, which in turn allocates some of that space to AT&T, etc.). Certificates are used to authenticate the validity of prefix advertisements. *Address Attestations* are delegator-signed statements that indicate an AS has the right to advertise a prefix (i.e., delegates to the AS).

Because of the costs associated with creation and validation (and to a lesser degree because of BGP message size constraints), the authors of S-BGP advise that address attestations should be managed through an out-of-band mechanism. The proposed architecture defines a collection of intermediate repositories maintaining certificates, revocation lists (CRLs), and address attestations. It is suggested that much of the effort of certificate and CRL vali-

dation can be completed by repositories. Centralized attestation repositories mitigate the costs of validation during *table resets* (i.e., memory re-initialization following a router reboot). For example, routers can rely on the repository to assert validity, rather than by validating received or acquired proofs. A challenge to S-BGP is the increased convergence time due to costs of signature validation [44].

One of the difficulties in the adopting any interdomain routing security solution is determining how it will integrate with existing infrastructure. In the Interdomain Routing Validation (IRV) project [15], participating ASes host servers called IRVs. Each IRV maintains a consistent corpus of routing data received and advertised. Remote entities (e.g., routers, other IRVs, applications) validate locally received data by querying source AS IRVs, using an out-of-band and potentially secure protocol. This approach has the advantage that the query responses can be tailored to the requester for optimization or access control, but more analysis of requirements and semantics are necessary to make the protocol fit for operational rigors.

The soBGP protocol combines proactive security measures with anomaly detection [11]. Like IRV, the proposed soBGP protocol focuses on incremental deployment [43]. soBGP validates address announcements in a similar manner to S-BGP address attestations. However, in an effort to make the solution more incrementally deployable, no authority (or structure of authorities) is mandated. Hence, users of the protocol are free to accept attestations or other routing policy data from any entity deemed trustworthy. Received policy data is used to identify and potentially discard suspicious BGP announcements through mechanisms such as RADIUS [32]. Because no structure of authorities is imposed, communities of soBGP ASes may quickly bootstrap and grow independently, but the number of configurable options in soBGP could potentially pose problems with interoperability [26].

An emerging protocol to comprehensive BGP security that combines attributes of S-BGP and soBGP is Pretty Secure BGP (psBGP) [58]. While soBGP can use a decentralized approach to authenticating public keys, in psBGP, the regional Internet registries (i.e., ARIN, RIPE, APNIC, LACNIC) act as certificate authorities that bind an AS number to a public key. Verification of address ownership, however, is performed in a decentralized manner. An AS creates lists of its AS number bound to the prefixes it owns, as well as the bindings of its AS peers, and distributes these through the network as a certificate, in a similar manner to soBGP.

The lack of origin authentication in BGP has led to the problem of multiple origin autonomous systems, or MOAS, which are discussed in [61]. Oscillating origins cause increased BGP traffic and can be traced to a relatively small number of prefixes [46], although the effect on traffic patterns is a subject of debate [1]. One approach to handling MOAS conflicts and other malicious BGP UPDATE messages models the AS topology, constructed with UPDATE messages, and passively monitors connections to compare announcements with the constructed connectivity graph in order to detect anomalous messages [31]. Another approach is to examine the routing tables of routers across the Internet, which can yield information on both potential MOAS situations and the address delegation hierarchy. Address allocation and its effects on routing table growth were studied in [7]. In addition, studies on the growth and evolution of routing tables have yielded models of address allocation for predicting the scalability of router memories [42] and table fragmentation [35]. However, neither of these works specifically focuses on the delegation hierarchy.

Whether by constructing and distributing cryptographic proofs or by detecting divergence from received policy data, the works described above acknowledge the importance of an address origin authentication. We begin our investigation of these issues in the following section by identifying a formal model of address management and considering the design space of origin authentication solutions, as well as the evolution of the address delegation hierarchy. We conclude in the latter sections by considering the applicability of these designs to the current Internet.

4. Origin authentication

Origin announcement authentication can be characterized by relations between organizations, ASes, and prefixes. The central goal of any address origin authentication solution is to provide evidence of these relations. Typically taking the form of cryptographically strong authentication tags, this evidence is used by receiving BGP speakers to validate address advertisements. The construction and use of these authentication tags is the topic of this work. We begin this section by formally

defining prefix announcements and ownership, and conditions for which valid assignments can be made. We then describe origin authentication tags, which are comprised of *delegation attestations*, whose constructions form the basis for simulation and evaluation later in this work. By providing formal definitions of the constructions, we can demonstrate their correctness and provide provable statements about their security. The constructions themselves are the tangible result of the formalizations of origin authentication that we derived, ensuring their validity. In later sections, we will discuss the practical implications of their use through simulations and analysis. This section concludes with discussions on expiration and revocation of attestations, as well as delegation proofs for aggregated prefixes.

4.1. Definitions and nomenclature

BGP address prefix announcements are essentially a pairing between an AS number and a prefix. The goal of origin authentication is to allow this pairing to be positively verified. Before describing origin authentication methods, we will first formally define AS numbers, prefixes, and BGP speaking organizations.

Let $\mathscr{ASN} = \{1, 2, \ldots, K\}$ be the set of all Autonomous System Numbers, where currently $K = 2^{16}$. Let $\mathscr{S}$ be the set of all BGP speaking organizations, i.e., those organizations to which AS numbers have been assigned by ICANN [25]. For each organization $C \in \mathscr{S}$, let $\mathscr{ASN}(C)$ be the set of AS numbers currently assigned to it. Let $\mathscr{O}$ be all of the organizations in $\mathscr{S}$ plus IANA and the other prefix registries. $\mathscr{O}$ is the set of all organizations that can "own" prefixes and may subsequently delegate ownership.

Since all prefixes are possible in an origin announcement, we take some care to define them and their structure below. Let $\mathscr{IPA} = \{0,1\}^{\ell}$ be the set of all ℓ-bit IP addresses where $\ell = 32$ for IPv4 and $\ell = 128$ for IPv6. Address prefixes, often just called prefixes, are denoted as x/j where $j \in \{0, 1, 2, \ldots, \ell\}$ and $x \in \{0,1\}^j$. Note that this is slightly different than the standard notation for prefixes n/j, where n is an ℓ bit long IP address and all of the $\ell - j$ least significant bits are assumed to be zero. For the remainder of this section we use the former, non-standard notation.

For the purposes of this discussion, an address range is a *set* consisting of the appropriate addresses. More precisely, $x/j = \{x \cdot y \mid y \in \{0,1\}^{\ell-j}\}$ which is simply all of the ℓ-bit addresses with the j most significant bits equal to x. (By convention, $\{0,1\}^0 = \emptyset$ (the empty set) so that $\emptyset/0 = \mathscr{IPA}$ is the set of all addresses. In firewalls, the set of all addresses is sometimes denoted as 0.0.0.0/0.) Using this notation, x/j is equal to the disjoint union of $x \cdot 0/(j+1)$ and $x \cdot 1/(j+1)$, where $a \cdot b$ represents the unary concatenation operator such that b is concatenated with a. Moreover, x/j is a superset of $x \cdot y/(j+k)$ for any $k \in \{0, \ldots, \ell - j\}$ and any $y \in \{0,1\}^k$. Note that the superset relation defines a partial order[3] on all address ranges. This partial order is naturally represented by a directed tree[4] where the root is $\emptyset/0 = \mathscr{IPA}$, where the leaves are the singleton sets w/ℓ and where the left and right child of x/j are $x \cdot 0/(j+1)$ and $x \cdot 1/(j+1)$, respectively. This tree is denoted the prefix tree. (For some purposes, it will be useful to extend this partial order to a natural total order, as we will see below.)

4.1.1. Delegation

The ownership of individual prefixes may be delegated from one organization to another several times. If an organization chooses to use a prefix of addresses under its ownership for its own hosts, rather than delegating the ownership of the prefix to another organization, it will assign that prefix of addresses to one of its ASes. The BGP speakers of that AS will then announce the pairing of that AS number with that prefix. For use below, we present a more formal description of a simple set of delegation and assignment options. Options that are more general are subsequently discussed.

For a given prefix y/k, an organization C may perform one or more of the following assignments or delegations:

1. $(y/k, n)$, where $n \in \mathscr{ASN}$, i.e., C assigns y/k to an AS number n;
2. $(y/k, C')$, where $C' \in \mathscr{O}$, i.e., C delegates y/k to C'';
3. $(y/k, \mathrm{R})$, i.e., C declares y/k as RESERVED.[5]

[3] More specifically, a lattice.

[4] Remove all partial orderings implied by transitivity and represent the remaining superset relations by a directed edge. This is the Hasse diagram of the partial order.

[5] RESERVED indicates that y/k should neither advertised nor delegated. We include this completeness, but for brevity defer further discussion.

The set of pairs is C's delegation policy for y/k.

C may be in error or it may attempt to cheat in several ways, and its delegation policy for y/k may thus be pathological. For example, $C \neq$ IANA may delegate y/k to another organization, even when no other organization had delegated y/k to it. C may delegate y/k to more than one other organization, or it may assign it to an AS number while also delegating it to another organization, perhaps mistakenly or maliciously. In these cases, its delegation policy consists of more than one pair. Below, we will enlarge the set of options available for a delegation policy to allow for incremental deployment. Before we do so, it will be helpful to define the delegation graph for y/k.

The delegation graph $G = (V, E)$ for y/k has a vertex set defined by $V = \mathcal{O} \cup \mathcal{ASN} \cup \{\mathrm{R}\} \cup \{\perp\}$. The set of edges E is defined as follows: for every organization C whose delegation policy for y/k is the empty set, a directed edge is placed between C and $\perp$. For every other organization D and every pair $(y/k, Z)$ in D's delegation policy for y/k, a directed edge is placed from D to Z where Z is in $\mathcal{O} \cup \mathcal{ASN} \cup \{\mathrm{R}\}$.

Definition. A node that has out degree of at least one but in degree 0 is called an *ownership source* in the delegation graph.

Note that IANA is an ownership source in the delegation graph of every prefix.

Definition. A node that has out degree zero but in degree of at least one is called an *assignment terminal* of the delegation graph. An edge into an assignment terminal is called an assignment edge.

Recall that by construction of the delegation graph, every node in $\mathcal{O}$ has at least one outgoing edge pointing to a node in $\mathcal{O} \cup \mathcal{ASN} \cup \{\mathrm{R}\} \cup \{\perp\}$. Thus, no node in $\mathcal{O}$ is a terminal.

Definition. An assignment edge is *ASN-respecting* if it is from an organization C to an AS number in $\mathcal{ASN}(C)$ or to R or to $\perp$.

Thus far, we have not constrained an organization's delegation policies for y/k in any way. Except for the fact that there are no terminals in $\mathcal{O}$, the delegation graph for y/k can be arbitrary. It can have multiple ownership sources, multiple assignment terminals, and multiple, intersecting paths. In fact, the delegation graph need not even be acyclic. Below, we define what paths in the delegation graph are valid, and then we will describe origin authentication tags, which can be used by those receiving BGP announcements to decide the validity of the delegation path among other things.

4.1.2. Validity of delegation paths

A path in the delegation graph for y/k is *valid* if

(a) the ownership source is IANA,
(b) the path is acyclic, and
(c) the assignment edge is ASN-respecting.

A partial delegation path, i.e., one in which the minimal node is in $\mathcal{O}$, is valid if the ownership source is IANA and the path is acyclic.

4.1.3. The acyclic requirement

The acyclic requirement for a valid path requires some discussion. A cycle in the delegation graph for y/k would seem to give each organization on the cycle equal claim to ownership to y/k and subsequent delegation or assignment. Clearly, an honest organization C would not purposefully participate in a cycle of delegation. The local connectivity of C in the delegation graph is not enough information to rule out being in a cycle when organizations that are not C's immediate neighbors are malicious or mistaken. In what we describe below when an organization C' delegates y/k to C, C' gives to C a set of delegation attestations,[6] one for each edge in the partial path. With these, C can determine the validity of the partial delegation path.

4.1.4. Null assignments

As defined, a valid path for y/k may have an assignment edge from C to $\perp$, which represents the fact that C's delegation policy for y/k is the empty set. This represents the following: when an organization has ownership of a large number of prefixes, it may never make BGP announcements for a large number of them. For example, several major backbone providers were delegated blocks of addresses of the form $x/8$ by IANA. They effectively own all of the prefixes that are subsets of their $x/8$, except for those they have further delegated. A provider's policy determines which of the subprefixes it will pair with which of its AS numbers in BGP UPDATE announcements, and which

[6] We adopt the term *attestation* from Kent et al. [29]. In the vernacular, attestations are proclamations of truth, and serve as good metaphors for statements of address delegation.

subprefixes it decides not to announce, at least until its policy changes. In practice, only a small fraction of the possible subprefixes actually appear in announcements (we establish this in Section 5).

4.1.5. Uniqueness

The definitions thus far do not rule out the possibility of a delegation graph that is a directed tree rooted at IANA, where every path is valid. To see this, consider the case where a valid partial delegation path ends in C, and suppose that C has received a proof of the validity of the path. Now suppose that C's delegation policy is of the form $\{(y/k, C'), (y/k, C'')\}$ where neither C' nor C'' are members of the original partial delegation path. From one valid partial delegation path ending in C, we get two valid partial delegation paths, one ending in C' and one in C''. Moreover, as we will see below, it is possible for C to construct a proof of validity of the partial path ending in C' and give it to C' and also to construct a proof of validity of the partial path ending in C'' and give it to C''.

Thus, a proof of validity of a delegation path is not sufficient to guarantee that the pairing of a prefix to an AS number in a BGP announcement is unique, or to guarantee that the organizations on the path have not been malicious or mistaken. To achieve this we require something more.

Definition. C's delegation policy for y/k is *faithful* as long as it consists of at most one pair. A path in the delegation graph for y/k is faithful if the delegation policy of every node on the graph is faithful.

Fact: There is at most one path in the delegation graph for y/k that is valid and faithful.

Thus, it is sufficient for receivers of announcements to check.

(a) the validity of the delegation path, and
(b) the faithfulness of the delegation policies of the organizations on the path.

We will discuss the former and the latter in turn below.

4.1.6. Incremental deployment

Before describing delegation attestations, we now describe a generalization of the above scheme that will facilitate incremental deployment. In addition to the three assignments or delegations listed above that C may perform for a given prefix y/k, an additional option is allowed:

4. $(y/k, \mathrm{U})$, i.e., C's delegation or assignment of y/k is UNAUTHENTICATED.

To describe the semantics of option 4, consider the delegation graph for y/k. Option 4 adds an edge from C to every node but C in V. The definition of a valid path remains exactly the same: the ownership source must be IANA, the path must be acyclic, and the edge assignment must be ASN-respecting. As before, C will compute and distribute a proof that $(y/k, \mathrm{U})$ is in its delegation policy for y/k. (It might put the proof in a public directory, such as those defined by S-BGP [53], where other organizations can obtain it.) Thus, it will still be possible for an organization to create a proof of validity for a valid path and for other organizations, i.e., those receiving the BGP announcement of a prefix, to verify the validity of the delegation path proof.

There are two primary reasons that C may declare y/k to be UNAUTHENTICATED. The first is that C has yet to complete any internal accounting and construction of proofs of which prefixes have been assigned to which of its own AS numbers. The second is that C has yet to complete its accounting and construction of proofs of which prefixes it has delegated to which customer organizations. In both cases, once an organization C has obtained the delegation for a set of prefixes, it will take some time to complete the accounting and construction of proofs. We will consider a generalization of the options above that allow C to restrict the set of possible next hops beyond the crude UNAUTHENTICATED option above in order to encode intermediate states of knowledge in its auditing and control process.

But, as C is going through this process, it may have intermediate states of knowledge. Thus, the delegation/assignments might include two additional options:

4a. C declares y/k as UNAUTHENTICATED but not delegated; and
4b. C declares y/k as UNAUTHENTICATED and delegated.

In the former case, a prefix in y/k may be paired with any AS number in $\mathscr{ASN}(C)$, and in the latter case, any organization not already on the partial delegation path may assume ownership of y/k. For simplicity, we suppress these options in the discussion below.

It is easy to see that having more than one node in a valid partial delegation path for y/k that has $(y/k, \mathrm{U})$

in its delegation policy does not increase the total number of valid origin announcements for y/k. To see this, consider a valid partial delegation path IANA, $C_1, \ldots, C_j$ where C_j is the first and only organization to declare y/k as UNAUTHENTICATED. The valid extensions of this partial path are either to an AS number in $\mathscr{ASN}(C_j)$ or to an organization in $\mathscr{O} - \{\text{IANA}, C_1, \ldots, C_j\}$. Let $\mathscr{E}$ be the set of all the AS numbers assigned to the organizations in $\mathscr{O} - \{\text{IANA}, C_1, \ldots, C_j\}$. Since C_j declared y/k as UNAUTHENTICATED, all origin announcements of the form $(y/k, n)$ where $n \in \mathscr{ASN}(C_j) \cup cE$ are valid. Lengthening the path further and allowing a downstream node declare y/k as UNAUTHENTICATED will not increase the set of valid origin announcements. Thus, for simplicity, and without loss of generality, we require a valid path to have at most one UNAUTHENTICATED declaration. Moreover, that declaration should be either by the last node in the path in $\mathscr{O}$ or be the second to last node in $\mathscr{O}$.

4.1.7. Faithfulness revisited

Before, we allowed declarations of UNAUTHENTICATED to be incorporated into the definition of a valid delegation path, requiring the delegation policies of the nodes on a valid delegation path to be faithful and restricting the number of valid delegation paths in a delegation graph to be at most one. Clearly, that is not the case when declarations of UNAUTHENTICATED are allowed on valid delegation paths. Nonetheless, without modification, the definition of faithfulness has meaningful semantics. Recall that the definition of a faithful delegation policy for a prefix is one that has at most one pair. If a delegation policy is not faithful, then an organization C may do the following. C may construct a delegation attestation of its declaration of UNAUTHENTICATED for y/k and pass that attestation to several organizations. C may also construct a delegation attestation for the delegation of y/k to C'. C' may not have knowledge of the attestation that C gave to other organizations. Of course, C will be constrained from behaving this way by economic incentives. Nonetheless, C' may appreciate the reassurance of a cryptographic proof of faithfulness. Moreover, those receiving origin announcements of y/k who have no direct economic relationship with C may find it useful when applying local policy to know definitively whether a prefix is provably UNAUTHENTICATED or has a unique, valid and faithful delegation path.

From the perspective of the delegation graph, the combination of faithfulness and UNAUTHENTICATED declarations yields the following.

Fact: For each terminal t in the delegation graph for y/k, there is at most one path between IANA and t that is valid and faithful. If no node on a valid and faithful path declares y/k as UNAUTHENTICATED then the path, and hence, the terminal, is unique.

4.2. Origin authentication tags and delegation attestations

In our scheme, origin announcements are verified by *origin authentication tags*, or OATs. An OAT consists of a delegation path, a set of *delegation attestations*—one for each edge in the path—and an *ASN ownership proof*. In order for an OAT to be positively verified, each delegation attestation must be positively verified, and the validity of the path must be verified. To check the validity of the path, it is simple to check whether the ownership source is IANA and whether the path is acyclic. To check whether the assignment edge is ASN respecting, the ASN ownership proof is used. To simplify, an ASN ownership proof is a statement signed by ICANN attesting to the fact that one or more AS numbers are among those granted to a particular organization. As with address prefixes, the chain of ownership/delegation may pass through more than one organization. The details of the ASN ownership proof are outside the scope of this paper. See the description of the S-BGP PKI [53] for a detailed description of one mechanism for ASN ownership proofs. As we will discuss below, OATs may accompany origin announcements or may be retrieved out-of-band by the receiver of an announcement, or part of an OAT may be retrieved in-band and part out-of-band, e.g., the ASN ownership proof.

In the previous section, we fixed a given prefix and considered every organization's policy for that prefix. Now let us fix the organization C and consider the collection of each of its delegations policies, one for each prefix. Let $\mathscr{D}(C)$ be the set of all prefixes such that C has a non-empty delegation policy for y/k. Assume for now that all of C's delegation policies are faithful. We will discuss this assumption further below.

Consider first delegation policies that represent delegations to another organization. If one of C's

delegation policies delegates x/j to C' then C has effectively delegated all prefixes that are subsets of x/j to C' as well. Thus, to minimize the number of explicit delegations, all parties in our scheme adopt the convention that explicit delegations from one organization have the *subtree closure property* defined as follows: if C explicitly delegates x/j to C' then C implicitly delegates all prefixes that are subsets of x/j to C'. Thus, since we are assuming faithfulness and the subtree closure property, if $x/j \in \mathscr{D}(C)$ is delegated to some organization C' then no prefix that is a strict subset of x/j is in $\mathscr{D}(C)$. Note that the encoding of prefixes as CIDR-blocks [14] ensures the subtree closure property.

For similar reasons, we adopt the encoding given by the subtree closure for the RESERVED and UNAUTHENTICATED declarations as well.

Now consider delegation policies that are assignments of prefixes to AS numbers. In this case, the subtree closure property is inappropriate. To see this, consider the following example in which C has been delegated the prefix x/j and all of its subprefixes by another organization. In addition, for simplicity, assume that C does not further delegate any of these prefixes to another organization. C may assign x/j to one of its AS numbers, say n_1. For many of the subprefixes of x/j, C may never make an origin announcement and thus, C's delegation policy for those prefixes is the null set. Moreover, C may assign a subprefix of x/j, say y/k, to another of its AS numbers, say n_2. To complete the example, suppose that all of C's delegation policies for subprefixes of y/k are null. The semantics of the longest prefix match encoding for routing tables means that the IP addresses in y/k will be routed to AS number n_2 and not AS number n_1. Note that origin authentication cannot defend against the attack that drops the $(y/k, n_2)$ origin announcement. The result of such an attack is that IP addresses in y/k are routed to AS n_1 rather than AS n_2. Such attacks are inherent to the longest prefix match heuristic.

To illustrate the definition of $\mathscr{D}(C)$, consider an honest organization C. C will only accept delegations of prefixes where the partial delegation paths are valid and where the delegation attestations are positively verified. Let $\mathscr{B}(C)$ be the set of prefixes explicitly delegated to C that meet these criteria. Since all such partial delegation paths are acyclic, this set is well defined. In this case, $\mathscr{D}(C) \subset \mathscr{B}(C)$.

4.3. Delegation attestations

We now describe three basic types of delegation attestations. For simplicity, we assume that an organization creates the same type of delegation attestation for each of its none-null delegation policies although in practice, it may implement a hybrid scheme. For all three schemes, we assume that the organizations creating the delegation attestations have public key signature keys and that the binding of these keys to identifying information of the organizations is given by certificate chains rooted by a CA with global BGP trust.

Before describing the basic schemes, we define the delegation function of an organization.

4.3.1. The delegation function

Since we are assuming faithfulness, C's delegation policies are equivalent to a function F_C with domain $\mathscr{D}(C)$ and range $\mathscr{O} \cup \mathscr{ASN} \cup \{\mathrm{R}\} \cup \{\mathrm{U}\} \cup \{\perp\}$. That is, for each $x/j \in \mathscr{D}(C)$, C's delegation policy for x/j is $\{(x/j, F_C(x/j))\}$.

4.3.2. Simple delegation attestation

The simplest type of delegation attestation for a prefix x/j is a signature by C of $(x/j, F_C(x/j))$, i.e., $[(x/j, F_C(x/j))]_C$ where the notation $[m]_C$ denotes m, σ where σ is the signature of m signed by C's key. Thus, if C uses only simple delegation attestations then we can write all of its delegation attestations as

$$[(x_1/j_1, F_C(x_1/j_1))]_C,$$
$$[(x_2/j_2, F_C(x_2/j_2))]_C,$$
$$\ldots$$
$$[(x_s/j_s, F_C(x_s/j_s))]_C,$$

where all of the prefixes of $\mathscr{D}(C)$ are represented.

Consider an example of an OAT for the origin announcement (12.1.1.0/24, AS29987) from Fig. 1 (except for the ASN ownership proof). The delegation path for 12.1.1.0/24 is (IANA, AT&T, ALPHA, AS29987). The delegation attestations for the path are

$$[(12.0.0.0/8, \mathrm{AT\&T})]_{\mathrm{IANA}},$$
$$[(12.1.1.0/24, \mathrm{ALPHA})]_{\mathrm{AT\&T}},$$
$$[(12.1.1.0/24, \mathrm{AS29987})]_{\mathrm{ALPHA}}.$$

Note that because of the subtree closure property for delegations, the first attestation that IANA delegated 12.0.0.0/8 to AT&T serves as an attestation that IANA delegated 12.1.1.0/24 to AT&T.

It is incumbent on the assumed certificate management infrastructure to issue and manage the identifiers. Note that in our design, unlike that of S-BGP [53], we allow the chain of delegations for address prefixes to be independent of the certificate chain for public keys. Organizations that may want to delegate address prefixes to other organizations may not want to operate as a public key certificate authority in order to do so. Of course, the semantics of the simple delegation attestations above can be included in certificates, which also serve to bind public keys to the originating and receiving organization names and address prefix as in [53]. The intent of our notation is simply to concentrate on the semantics of the delegation path, rather than on the details of the PKI.

These simple delegation attestations are easy to construct, maintain and distribute. However, because each association must be created (signed) and validated individually, they can place significant resource burdens on the routes of both the issuing and verifying organizations [28] (see Section 6 for further analysis).

4.3.3. Authenticated delegation list

To reduce the cost of signature creation and verification required by simple delegation attestations, an organization can create a single list of all of its delegations and sign that list. Such a scheme could be written as

$$[\quad (x_1/j_1,\ F_C(x_1/j_1)),\ (x_2/j_2,\ F_C(x_2/j_2)),\ \ldots,\ (x_s/j_s,\ F_C(x_s/j_s))\quad]_C,$$

where $\mathscr{D}(C) = \{x_1/j_1, \ldots, x_s/j_s\}$.

For each origin announcement received by a BGP speaker, that speaker must acquire the authenticated delegation list of every organization on the delegation path, in order to positively verify the pairing of the prefix to the AS number. Clearly, the authenticated delegation lists of some organizations may be quite large. Hence, verifiers must commit significant bandwidth and storage. However, the computational costs of verifying a large number of simple delegation attestations are largely avoided. The efficacy of authenticated delegation lists is evaluated experimentally below and compared to that of simple delegation attestations.

Of course, the authenticated delegation list and the simple delegation attestations are two extremes in a spectrum of possibilities. Rather than signing the entire list, an organization may break up the entire list into several lists and sign each of the smaller lists. A natural means of breaking up the list is according to those prefixes that are delegated to the same organization or assigned to the same AS number (called an *AS authenticated delegation list*). This latter design most closely resembles the address delegation certificates of S-BGP [29]. The advantage of this approach is that the AS can collect proofs for all addresses that it originates. These proofs can be distributed by the AS upon request or in conjunction or within `UPDATE` messages. We explore this and other operational considerations in Section 6.

4.3.4. Authenticated delegation tree

Consider the following scheme. An organization C creates a Merkle hash tree [36]. The values of the leaves of the tree are of the form $(x/j, F_C(x/j))$ for each $x/j \in \mathscr{D}(C)$. The value of each internal node of the tree is a hash of the values of the children of the node. We assume that the hash function used to create the hash tree is collision resistant. Let h_0 denote the value of the root. C signs the root, $[h_0]_C$. Because of the efficiencies afforded by their construction, Merkle hash trees are widely used in security (e.g., for BGP path verification [19]).

In this scheme, the delegation attestation that C is delegating/assigning x/j to $F(x/j)$ consists of the value of the siblings of all of the nodes on the path in the Merkle tree from $(x/j, F_C(x/j))$ to the root plus $[h_0]_C$. This is sufficient information for a receiver to recompute the hash values along the path from $(x/j, F_C(x/j))$ to the root, check that it is equal to h_0, and verify C's signature on h_0. The size of a single proof is logarithmic in the size of $\mathscr{D}(C)$. Because prefix tree proofs share intermediate nodes, the distribution costs can be amortized.

It is easy to see that if an adversary is able to create a delegation attestation for a pair $(x/j, Z)$ that is not one of the leaves of C's authenticated delegation tree, then it has either found a collision of the hash function or forged a signature. Since both are assumed to be infeasible, creating bogus delegation attestations for authenticated delegation trees is infeasible.

4.3.5. Authenticated delegation dictionaries

Naor and Nissim introduced the notion of authenticated dictionaries [41] that in our context is useful for enforcing faithfulness, as we will see below. The model for an authenticated dictionary

is that a user may make queries to a directory asking whether an element of the universe is in the dictionary (which is a subset of the universe). The dictionary owner gives the directory sufficient information for the directory to return yes or no along with a proof in either case. Since a valid proof is required for both membership and non-membership, the directory is forced to answer correctly. In addition, the authenticated dictionaries in [41] have the property that they are efficient to update.

In this paper, we define an authenticated delegation dictionary for an organization. This is simply an authenticated dictionary where the elements of the dictionary are the elements $(y/k, F_C(y/k))$ for each $y/k \in \mathscr{D}(C)$. To make this concrete we briefly describe the scheme in [41] modified to this context.

We start with a search tree in which the leaves are sorted, say, left to right, based on the search key. For the sake of efficiency [41], we use 2–3 trees. In our case, the search key will be the address prefixes. We have already described the natural partial order of the prefixes whose Hasse diagram is a tree. We define an extension of this partial order to a total order defined by a prefix's position in the depth first search of the entire prefix tree. Note that this total order respects the partial order. It is easy to see that this order is essentially a lexicographical ordering of the prefixes. That is, the order can be described by the relations

$$x/j < x \cdot y/(j+k) < z/j$$

for any $j \geqslant 0$ and $k \geqslant 0$ respecting $0 \leqslant j+k \leqslant \ell$, and any $y \in \{0,1\}^k$ and any x and z in $\{0,1\}^j$ with $x < z$. As an example, all of the address prefixes of a subtree rooted at x/j appear consecutively, with the smallest element being x/j itself, and the largest element being the rightmost leaf of the subtree $x \cdot 1^{\ell-j}/\ell$.

In the ADD for C, we build a balanced 2–3 search tree where the leaves are of the form $(y/k, F_C(y/k))$ for each $y/k \in \mathscr{D}(C)$, sorted according to y/k. We augment this tree as follows. The value of an internal node is the concatenation of the search tree keys of the node and a hash of the values of all the child nodes. The root of the tree is signed by C. A delegation attestation for $(y/k, F_C(y/k))$ consists of the signed root, the search tree path from $(y/k, F_C(y/k))$ to the root, and the value of the children of the nodes of the path.

Recall that if the delegation policy for y/k is the empty set, then y/k is not a leaf of the ADD. A proof to that effect consists of a positive proof, as above, for the largest leaf key smaller than y/k and a positive proof of the smallest leaf key larger than y/k. Positive path proofs for both of these elements can be used to verify that they are consecutive leaves in the sorted order. Also, recall that if y/k is delegated to C' then by the subtree closure property, all of the delegation policies of the proper subprefixes of y/k should be empty. That is, none of the proper subprefixes of y/k should be in the ADD. A proof to that effect consists of a positive proof of the leaf with key y/k and a positive proof of the smallest leaf key larger than y/k. This leaf key must be larger than $x \cdot 1^{\ell-j}/\ell$ in order to provide a proof that C has been faithful for all subprefixes of y/k.

Note that an organization can give an ADD to a directory and the directory can verify the construction of the tree and signature on the root (actually the organization need only give the leaves of the tree and the signature of the root and the directory can rebuild the tree and verify the signature). In particular, the directory can check that no two leaves have the same key. As discussed earlier, to guarantee that multiple ASes are not announcing the same address prefix (in the case where UNAUTHENTICATED is not on the delegation path) it is sufficient to check that the delegation policy of every node on the path is faithful. Checking the faithfulness of an organization's delegation policy can be done if the organization places its authenticated delegation dictionary in a directory such as the ones proposed in S-BGP [53]. The proof of faithfulness of a delegation policy must be placed in a publicly queriable repository; otherwise, an organization can reply with different proofs of its own making to different entities.

An advantage of a 2–3 tree over other structures (e.g., binary tree) is in the cost of updates. Adding or removing associations from a binary tree may require balancing, which can affect a large number of intermediate nodes. Communicating these new values to the potentially large number verifier routers can be costly. The likelihood that the tree needs balancing (and hence the cost of communicating updates) is lower in a 2–3 tree scheme. The initial communication costs in a 2–3 scheme are higher than in a binary tree scheme (because of the additional sibling nodes). Hence, the best approach scheme for a given environment is determined by the number and frequency of updates. We investigate the stability of assignments and evaluate the costs of these schemes using real BGP trace data in Section 5.

4.4. Expiration and revocation

As with any system involving public key signatures and certificates, there are a host of issues involving protection from replay, expiration, revocation, etc. For simplicity, we did not explicitly include an expiration time in our description of delegation attestations, but in any actual operational implementation, an expiration time would be included. In many cases, the prefix delegation involves a customer/provider relationship. For example, either the provider delegates one of its prefixes to a customer, or the customer owns an address prefix and delegates it to the provider, as displayed in Fig. 1. In these cases, the expiration in the delegation attestation would naturally be set to the expiration date of the customer/provider service agreement.

Replay protection can easily be achieved if delegation attestations are retrieved out-of-band by verifiers over a secure channel (e.g., TLS) from a directory. In-band deliveries of delegation attestations are susceptible to replay attacks (e.g., C announces a prefix, and then withdraws it, whereupon C' replays the original announcement along with the original OAT that has not expired). Our scheme can be augmented to require short-lived "liveness" tokens such as those in [38,4] that have very short durations, e.g., good for one day, while the delegation attestation can continue to have a longer duration. In such systems, both the delegation attestation and the liveness token need to be positively verified. As always, there is a tradeoff between administrative and computational overhead and reducing the period of vulnerability. Because this information never expires, it may be held for months or years. Conversely, traditional routing protocols (e.g., RIP, OSPF) ensure route consistency refresh: routing data periodically expires and is reasserted. The original designers of BGP discarded the refresh model as a candidate approach largely because it was assumed that it would not scale to interdomain routing [17]. The semantics of BGP are in conflict with the natural security requirements of origin authentication. For example, all updates are passively received and hence are attractive targets of denial of service attacks. If an adversary can prevent delivery of an announcement, the verifier will never learn of it. Where the announcement supersedes an existing one, the lifetime of the invalidated proof is extended indefinitely.

Good security practices mandate that OA proofs have an explicit lifetime over which they should be considered valid. Because proofs (and by indirection prefixes) expire, new proofs of their continued validity must be obtained. We envision two possible extensions to BGP for this purpose:

- *REASSERT message*—An additional REASSERT message would be added to BGP. This statement reasserts the validity of a prefix announcement via (a new) proof. The frequency of REASSERT messages would be significantly slower (e.g., could be on the order of days or weeks) than the refresh messages in more traditional protocols. Because of their low frequency, the costs of the additional messages would be nominal.
- *Off-line verification*—Received updates are validated by external entities. Speakers acquire validation information administratively or using external protocols from a representative of the originating organization or AS (e.g., S-BGP repositories [29], IRVs [15]). Entire trusted domains (e.g., ASes, confederations) can consolidate verification by centralizing and distributing validity information to each BGP speaker. This latter approach can dramatically reduce the burden of OA on a domain.

Revocation is motivated by administrative or security concerns. Administrative revocation occurs when a prefix is migrated from one AS or organization to another because of a change in topology or affiliation. Revocation due to security occurs where some AS is found to behave in an inappropriate or malicious manner. In this case, revocation removes the rogue entity's right to act as the origin of a prefix.

Administrative revocation is semantically similar to route withdraws in BGP: an originating AS wishes to announce that it is no longer serving the prefix. The AS is expected to act in accordance with the protocol by withdrawing the route. Conversely, security related revocation is much more dynamic, and reflects active attempts to circumvent the correct operation of the network. Hence, the threat model under which the security relevant revocation should be considered is significantly stronger.

One argument asserts that no additional infrastructure is needed for administrative revocation. The AS which loses control of a prefix immediately withdraws the associated route (and will not

announce it further). The new origin AS receives a new proof from the delegating entity and can announce it freely. The prefix is attributed to the new AS through the normal routing protocol as bounded by the route propagation delay.

Revocation due to malicious behavior is much more difficult. Guaranteeing that a particular proof has or has not been revoked is analogous to the widely debated problem of certificate revocation. Note that such events are extremely rare: we were unable to discover even one occurrence where an address space removed from an AS in response to malicious behavior. The limited frequency and relative difficulty of the problem lead us to believe that the risks are far outweighed by the costs of a comprehensive revocation solution. However, if it is deemed necessary, existing approaches to certificate revocation can be used [38,30,41,18,34].

4.5. Aggregation

Aggregation allows an AS to encapsulate a set received prefixes in a single `UPDATE` message (with a shorter address/mask that completely encompasses the received prefixes). This is used where the set of common prefixes is advertised to the network through a single AS path passing through the aggregating AS. In this sense, aggregation allows an AS to assume the role of origin for a set of common prefixes. This greatly enhances the scalability of BGP by reducing the state held at each router. Note that aggregation involves the confluence of both the prefix delegation graph and network topology.

Our framework naturally allows for aggregation. Consider the following example. Organization C delegates $y \cdot 0/(j+1)$ to C' and $y \cdot 1/(j+1)$ to C''. In addition, it assigns y/j to one of its ASes numbered n. Also suppose that the ASes of C' and C'' are downstream of AS n in the network topology. Of course, C' and C'' can make origin announcements with valid OATs for prefixes or sub-prefixes of $y \cdot 0/(j+1)$ and $y \cdot 1/(j+1)$, respectively, and those announcements need not go through AS n (e.g., due to multi-homing). But those announcements that do go through n can be aggregated by AS n that can send out an announcement for $(y/j, n)$ with a valid OAT. A slightly larger set of aggregation alternatives for C is possible using the generalizations to our scheme discussed in the following subsection.

A simple extension supporting aggregation would have each owner generate a *delegation proof* stating that the upstream AS has the right to aggregate the prefix (e.g., by including all aggregating ASes in the node value). These proofs would be distributed as others are. The validator must check all the prefix proofs, rather than validating the single received prefix. Hence, message size, state, validation costs would grow linearly in the number of encapsulated addresses. This potentially nullifies the advantages of aggregation.

Note that aggregation is an issue of network topology, rather than of prefix ownership. These topological relationships are more fluid than address ownership. Hence, it may not be appropriate or desirable for the ownership source to assert control over aggregation. The generation of delegation proofs required by aggregation calls for a more general routing security framework.

4.6. Generalizations

There are number of natural generalizations to the above scheme. Consider the following delegation option for an organization C for an address prefix y/k:

1′. $(y/k, \mathscr{C}, \mathscr{N})$ where $\mathscr{C} \subset \mathscr{O}$ and $\mathscr{N} \subset \mathscr{ASN}$.

All the previous options can be captured with this as follows. Option 1, the ASN assignment option, is given by $|\mathscr{N}| = 1$ and $\mathscr{C} = \emptyset$. Option 2, the delegation option, is given by $|\mathscr{C}| = 1$ and $\mathscr{N} = \emptyset$. Option 3, the RESERVED option, is given by $|\mathscr{C}| = \mathscr{N} = \emptyset$. Option 4, the UNAUTHENTICATED option, is given by $|\mathscr{C}| = \mathscr{ASN}$ and $\mathscr{N} = \mathscr{O}$. The semantics of this option in terms of the delegation graph are similar to those described for the UNAUTHENTICATED option above except that rather than adding edges between C and all of the nodes of the delegation graph, edges are added between C and the nodes of $\mathscr{C}$ and $\mathscr{N}$. The option is meant to capture the case in which an organization has not completed its audit of certain parts of its address space but it has narrowed down the possibilities for certain address blocks. For example, it may wish to encode in an attestation that only some subset of its customers can legally be the next hop in the prefix delegation path.

A more general delegation option still for C is

1″. $(y/k, \mathscr{Q})$ where $\mathscr{Q}$ is a subset of all possible paths in the delegation graph from C.

Essentially option 1′ is a way for C to describe and restrict all of the possible next hops. However,

C may wish to impose further restrictions beyond the next hop. In particular it may wish to delegate y/k to another organization C' but not allow C' to delegate the address prefix further (i.e., require C' to assign y/k to an AS number).

The definitions of the validity and faithfulness of a path are easily extended to cover these more general cases. Efficient encodings for these options and other issues are deferred for future consideration.

5. The address delegation graph

The cost of origin authentication systems in general, and of the constructions defined in the preceding sections in particular, are a reflection of prefix reference locality and delegations of the address space. Any evaluation of an OA must be based on a firm understanding of these factors. Address reference locality is easily ascertained from publicly available BGP update streams. Conversely, due to the difficulty of determining the exact delegation structure, we estimate the *address delegation graph* of the IPv4 address space. This graph is further used as input to our simulations of OA services in Section 6.

Before attempting to reconstruct the graph for the current Internet, we reiterate the intuition behind its structure. The delegation graph is a directed graph of organization-to-organization delegations of address space. As formally defined in the preceding section, IANA is the root node, and all the registries, ISPs, and other organizations form child nodes. Every delegation of address space is reflected in a single edge in the graph. For example, an edge would be added when IANA assigns a block to RIPE or when AT&T hands off a block of address space to a customer. The graph resembles a tree because delegations largely flow from IANA, to registries, ISPs, and ultimately end-customers. The graph illustrates how each address block was assigned to the AS that originates it, and ultimately shows the set of delegations that must be validated to achieve secure address use in the Internet.

5.1. Approximating IP address delegation

While previous studies have accurately reconstructed the routing topology graph [56], it is exceptionally difficult to approximate a delegation graph for the Internet. To show why this is so, consider the fragmentation of AT&Ts `12.0.0.0/8` address space. A recent evaluation of BGP updates for a single day showed 571 different ASes announced 923 distinct prefixes in the `12.0.0.0/8` range.[7] The delegation of these prefixes often occurred years ago. Moreover, many organizations to which address space was delegated no longer exist, have changed hands, or currently have no formal relationship with AT&T. Hence, reconstructing and recording these delegations would be an arduous process. Doing so for every organization in the Internet may take years. For this reason, any solution must be incrementally deployable: we as a community simply cannot wait for all delegation to be discovered and recorded.

In a related work, Kent et al. estimated the statistical properties of the IPv4 address delegation while investigating deployment costs of S-BGP [28]. They determined the number of delegated address, organizations, and ASes using Merit BGP statistics and other public data as of February 1999. As was appropriate for their purposes, this work only estimated the size of the problem, but did not consider its structure. It is this latter feature that is most relevant to the current work; we wish to understand the how and by whom delegation occurs. We also found the statistical properties of BGP have shifted significantly since the original study. For example, we identified a BGP speaker who received 300 times the number of UPDATEs cited in the previous study (1500 in 1999 versus 600,000 in 2003). This differential may be partially explained by the original study filtering iBGP (we did not). Note that we seek solutions that can sustain the worst-case load, and we therefore focus on the largest visible load on any one BGP speaker. The ratio of iBGP to eBGP traffic is topology dependent and highly variable. However, we wish to measure the worst case (as it serves as the limit) and hence consider a heavily loaded environment.

In recognition of the problems inherent in determining a perfect representation, we approximate the delegation graph using available interdomain routing information. The following lists several of the relevant sources and considers the quality of delegation information that they represent.

(a) *IANA*—IANA is the origin of all delegation of IP address space. IANA directly delegates address space to 46 unique organizations

[7] This figure includes prefixes delegated out of AT&Ts address space, as well as a fraction of the prefixes multi-homed by AT&T.

[23]. These delegations show the broad allocation of address space on the Internet, and must be incorporated into any approximation of the graph.

(b) *BGP announcements*—One can estimate delegation by looking at announcement encapsulation. Assume that all ASes announce every prefix they are delegated. Any advertisement *encapsulated* (i.e., has a longer matching prefix) that is from another AS could be considered legal delegation. Note that this may be a very good predictor of address space delegation; every delegation found by this method represents at least one legal delegation (because no legal delegation will give the same range to two different ASes).

(c) *AS topology*—Historically, many organizations have received address space from their connectivity providers. This organizational linkage is often reflected in the AS topology. Hence, the AS topology can contain partial information about the address space delegation.

We note that some parts of the delegation graph can only be discovered by communicating with the parties involved. Some organizations, most notably IANA, own parts of the address space but do not directly participate in BGP. Hence, the accuracy of any approximation is partially dependent on the degree to which this information is public. In general, approximations arrived at using the above methods are almost certainly going to underestimate the number of delegations (because of these unexposed organizational relationships). Our intuition and anecdotal evidence suggests that such relationships represent but a small percentage of total delegations. However, we do consider the possible effect of underestimation on our results in Section 6.3.

5.2. An approximate graph

We have selected IANA and BGP announcements to approximate the delegation graph. We chose not to use the AS topology information because it was unclear how such information could be rationally interpreted with respect to delegation. While topology information reflects current relationships, IP address assignments often represent delegations that occurred long ago. Moreover, many, if not all of the relations between organizations that would be used to inform delegation are reflected in the BGP announcements. The RouteViews project [37] repository is our source of BGP announcement data. The delegation graph integrates public information published by IANA, obtained as a single table update from February 15, 2005. The BGP table contained 169,459 distinct prefixes advertised by 19,044 ASes. Such numbers are consistent with Huston's detailed ongoing evaluations of BGP advertisements [21].

One of the challenges in constructing an approximate graph was making connections between the IANA (organizational) and BGP announcements. In looking at the BGP data, we found several prefixes handed out by IANA had a single corresponding AS announcement. For example, we found that the AS 7018 advertised `12.0.0.0/8`. Not surprisingly, 7018 is one of the ASes owned by AT&T. This is an assignment from the AT&T organization to its own AS. We added an assignment edge to the graph for each such announcement using IANA supplied Organization to AS bindings [22]. All other non-self delegations were handled in a similar manner; a delegation edge was added from the parent organization where no encompassing AS advertisement existed. In the absence of other information, dummy organizations were added for each AS. This graph construction process is illustrated for a small part of the address space (`12.1.0.0`) in Fig. 2.

Several kinds of UPDATE announcements were not useful in generating the graph. UPDATES representing self-deaggregation were not useful. Self-deaggregation occurs when an AS announces a prefix completely encompassed by another prefix announced by that same AS (e.g., if one of AT&Ts AS announced both `12.0.0.0/8` and `12.1.0.0/16`). These longer prefixes were ignored.

The delegation graph used in this section was generated in the following way:

1. We create a node for every organization in the IANA address delegation list published on its website on February 20, 2005. We associate the advertised prefixes listed in those delegations with each node.
2. We create a node for every AS listed in a single Routeviews route table collected on February 15, 2005. We associated every prefix listed in the table with the node of the originating AS.
3. For every prefix: create an edge from a prefixes node to the node of its *closest parent*. The closest parent is the longest prefix that completely sub-

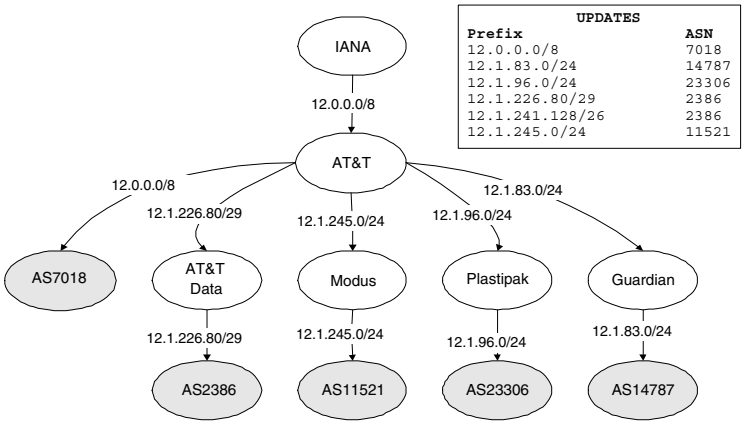

Fig. 2. Address delegation graph for prefix 12.1.0.0/16.

sumes the prefix. For example, assume that the following three advertisements were observed: 12.0.0.0/8, 12.1.0.0/16, and 12.1.2.0/24. In this case 12.0.0.0/8 would be the parent of 12.1.0.0/16 and 12.1.0.0/16 would be the parent of 12.1.2.0/24. The intuition here is that any AS advertising a child prefix must have (directly or indirectly) received that address space from its parent. Hence, we identify that as a delegation of address space.

The graphs used in the remaining sections were generated in an analogous way, save that the source data may have been taken from different dates. We explicitly identify the source data dates where they differ throughout.

The approximated graph shows that 2112 of out of 14,912 total organizations delegate prefixes to other organizations. This seemingly small number of address delegating organizations is consistent with the growth of the Internet: address space has largely been handed out by providers to customer organizations. Customers do not frequently further delegate received address space to others. Interestingly, the IANA and BGP data led to only 114,183 delegations and assignments requiring proofs.[8]

In Fig. 3 we rank each node according to the number of delegations from that node in the delegation graph, then plot the number of delegations versus rank. When viewed on a log–log scale, the plot is essentially linear and hence conforms to the classical Zipf distribution [62]. (In addition to conforming to a Zipf distribution, the delegation structure also follows a power law. That is, the number of nodes $n(d)$ that each have d delegations from that node versus d is given by $n(d) \sim 1/d^{\beta}$ for some constant β [3,6,12]. The power law delegation distribution implies the Zipf distribution for number of delegations.) The most striking fact shown by this data is that the overwhelming number of delegations are being performed by relatively few ASes/organizations. In this case, 16 ASes/organizations are responsible for 80% of the delegation on the Internet. Furthermore, 122 ASes/organizations are responsible for 90% of the delegations and 1220 perform 99% of the delegations. The top 10 delegators are: 1-ARIN (30%), 2-*various registries*[9] (15%), 3-APNIC (12%), 4-RIPE NCC (8%), 5-RIPE (4%), 6-LACNIC (3%), 7-AT&T (2%), 8-UUNET (1%), 9-ARIN Cable (1%), and Sprint (1%).

The small number and delegation densities indicated by this study shows that the proof system approaches identified in the preceding sections are likely to be advantageous. Proof systems (i.e., delegation trees, delegation lists) improve performance where few authorities provide proofs to arbitrary collections of constituents. We revisit and confirm this via simulation in Section 6.2.

[8] We found many prefixes that did not require any origin proof. For example, any prefix that deaggregates a prefix owned by the same organization does not require a proof.

[9] IANA has delegated several blocks of address space to an unspecified collection of registries. This block was modeled as a single delegator for the purpose of this analysis, and is likely to be spread out over the various address registries (e.g., RIPE, etc.). The proper attribution of this space would likely increase the "market share" of the cited registries and hence further increase the approximated delegation densities.

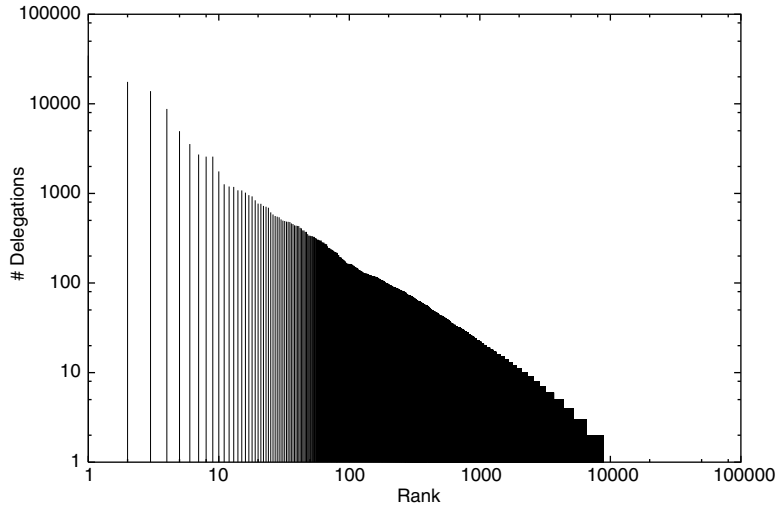

Fig. 3. Delegation—cumulative distribution function for the delegation in the approximate delegation graph.

5.3. Delegation stability

The stability of the delegation hierarchy contributes greatly to the performance of origin authentication. If delegation is highly fluid, then it will be difficult to efficiently construct and distribute the rapidly changing proofs. In general, routing data has been shown to be surprisingly stable [51]. This section considers if the same is true of the delegation of the IPv4 address space. Note that this study serves as a starting point of a larger effort. We are currently studying origin change inter-arrival times in conjunction with other artifacts of BGP traffic in an effort more firmly establishing address churn in inter-domain routing.

Table 1 depicts the stability of IP address delegation over first five months of 2003. We obtained a single BGP table from the first day of each month from the RouteViews repository. The table data is used to approximate the Internet delegation hierarchy (using the algorithm defined above) on each day. The table shows the measured change between each consecutive month (e.g., January–February) and over the entire period (e.g., comparing January–May). A delegation is *added* when it appears in the hierarchy for the second month but not the first, *removed* when it appears in the first but not the second, *moves* when the originator changes, and is *stable* when no change is observed; *total* reflects the number of unique delegations.

These first five columns of the table represent a worst-case analysis: the number of adds and removes may be overestimated because some prefixes are not present in the table during the recorded periods (because of transient network issues). Similarly, legitimate moves cannot be differentiated from MOASes or prefix hijacking. Hence, we can say that the delegation is no more unstable than is indicated by this analysis.

We approximate best-case stability by filtering all suspicious adds, removes, and moves. An event is deemed suspicious if it occurs more than once for a prefix. For example, if a prefix is marked as moving more than once, it is likely that it is oscillating between ASes (e.g., due to multi-homing). Because the move does not represent a new delegation of address space, it can be ignored for the purposes of this analysis. Of course, this approximation is still imperfect; we cannot differentiate a legitimate move from a multi-homed prefix that only oscillates between ASes once in our test data.

Moves are the most disruptive operation. A legitimate *move* indicates that a part of the address space has been revoked from one organization or AS and subsequently delegated to another. Both revocation information and proof updates must be distributed throughout the network. All month-to-month comparisons show a very small number of moves (ranging from 1.1% to 1.9% in the worst case, and .5% in the approximate best case).

Adds and removes are less urgent. Because they do not affect currently advertised routes (in the case of adds) or do not require immediate revocation (in the case of removes), some notification latency is acceptable. The number of adds and removes in any given month is relatively small (3.1–7.2%). This

Table 1
Delegation stability—worst case stability of the IP address delegation graph from January to May 2003

Class	January–February	February–March	March–April	April–May	January–May	January–May (filtered)
Stable	117,117 (90.0%)	116,741 (90.1%)	116,340 (87.5%)	119,701 (89.0%)	103,397 (72%)	128,350 (89.6%)
Added	5774 (4.4%)	4925 (3.8%)	9667 (7.2%)	5800 (4.3%)	19,001 (13.2%)	6977 (4.8%)
Removed	5465 (4.2%)	6207 (4.7%)	4246 (3.1%)	7017 (5.2%)	15,770 (11.0%)	7052 (4.9%)
Moved	1632 (1.1%)	1575 (1.2%)	2655 (1.9%)	1944 (1.4%)	5047 (3.5%)	836 (0.5%)
Total	129,988 (100%)	129,448 (100%)	132,908 (100%)	134,462 (100%)	143,215 (100%)	143,215 (100%)

The filtered data approximates best-case stability of the delegation graph (Figs. 4 and 5).

indicates that the delegation hierarchy evolves slowly, where only about 10% of the delegations (representing 10–15 thousand delegations in the worst case) change on any given month. Moreover, as shown by the January–May measurements, the rate of change is relatively constant. The best-case analysis exhibits similar properties, albeit at about half the rate of change.

We extend these results to cover the 30-month period from July 2002 to December 2004 in order to determine overall trends, as shown in Fig. 4. This graph shows that the number of prefix delegations is increasing with time, and more prefixes are being announced than removed. We also see that in keeping with the results determined from the previous table, the number of moves is extremely small compared to the number of stable delegations.

Next, we consider weighting the delegations in order to get a better sense of the true effects on the address space. If, for example, delegations are predominantly comprised of small blocks, they will have minimal effect on the overall delegation hierarchy. To this end, we consider the delegation in terms of /24 address blocks, equivalent to 256 IP addresses (the size of the traditional "Class C" delegation). The /24 block is the smallest that is likely to pass the BGP filter of many organizations, so we discard delegations of smaller blocks than this.[10] The results of the weighted delegations are shown in Fig. 5. The percentage of delegated address space that is stably delegated is even higher than in the previous case, with the majority of months registering between 91% and 97% of delegated space as stable compared to the previous month, and the amount of address space moved is less than 2.5% for 24 of the 30 months considered.

[10] In many transit ASes, address blocks smaller than /19 were traditionally filtered, but because of the need for finer-grained delegations due to factors such as multihomed organizations, this is changing.

Note that there is a large anomaly in the weighted graph, where the amount of delegated space suddenly jumps to twice the previous size before subsiding three months later. This large disturbance warranted further investigation. The data we examined came from the full corpus of Routeviews data, with 59 listeners contributing routing tables to the repository over the 30-month span. From all these routers, we determined that one peer started advertising delegations for a large portion of address spaced reserved by IANA in May of 2003. This routing peer, `194.85.4.55`, traces to RUSnet, an ISP based in St. Petersburg, Russia, and the advertisements showed these reserved address blocks delegated to AS 3246, which belongs to TDC Song Networks, a Scandinavian ISP. Curiously, no other Routeviews peer advertised these routes, likely a result of an appropriate BGP filtering policy put into place by the respective organizations. However, the following month shows that a large part of this newly advertised space has moved, an even more anomalous situation. We found that these routes were no longer being advertised by RUSnet, but were instead being advertised by `196.7.106.245`, belonging to UUNET South Africa, based in Cape Town, South Africa. The reserved space was no longer being delegated to AS 3246, but was now being advertised as belonging to AS 2905, owned by UUNET South Africa. This was advertised for a month before being removed from the routing tables. Similarly in this instance, no other Routeviews peer advertised these routes. No discussion of these events was found in the NANOG (North American Network Operators Group) or RIPE discussion lists, and no explanation has been proffered as to why these delegations occurred [50]. Our assumption is that these were the result of errors in configuring policy that were corrected later. Generally, announcements of newly advertised space are posted to NANOG and RIPE mailing

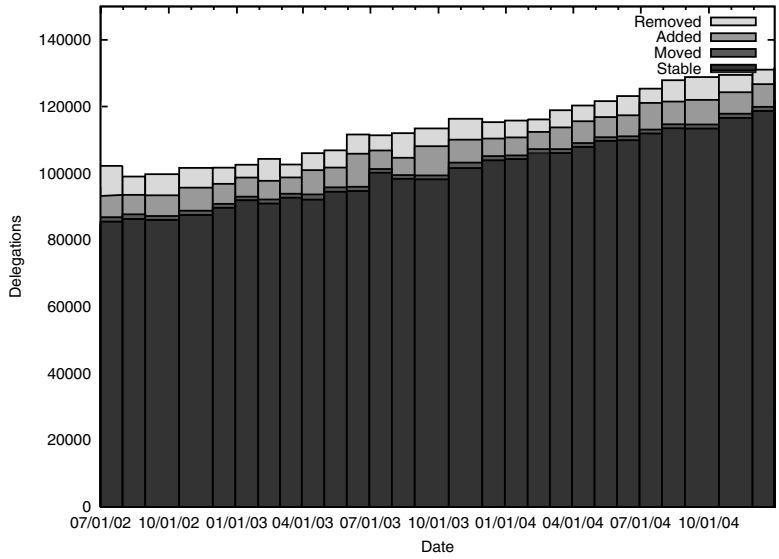

Fig. 4. Worst-case stability of the delegation graph from July 2002 to December 2004, by number of delegated prefixes.

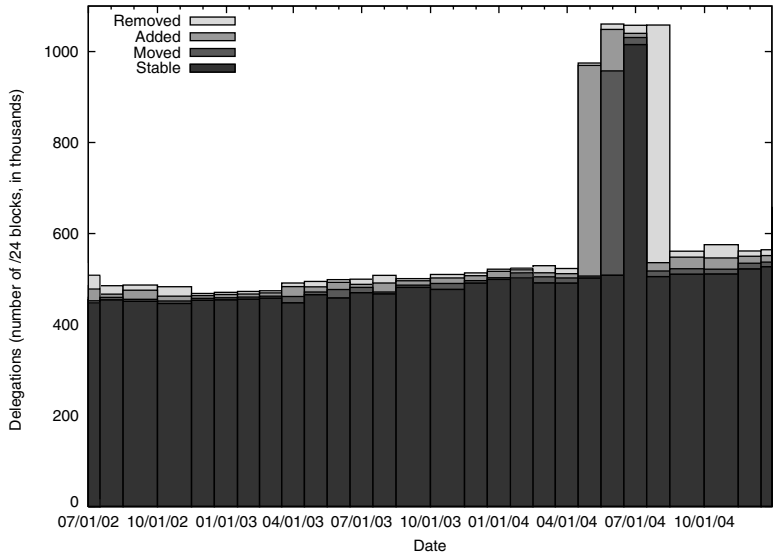

Fig. 5. Worst-case stability of the delegation graph from July 2002 to December 2004, weighted by the number of affected prefixes.

lists, where network operators will see to reconfigure their BGP filters accordingly; because these reserved blocks were not announced for advertisement, these delegation errors may have passed undetected by other organizations. This may be evidence of a validity fault as detailed in [13]. When these erroneous delegations are removed, we find less than 1.5% of prefixes have moved and over 94% of delegations remained stable, which matches the results garnered from other months as demonstrated in the graph.

6. Evaluation

The previous sections, particularly Section 4, formally modeled a number of approaches to origin authentication, culminating in four unique constructions that can be used to provide attestations of address ownership. Each of these approaches has unique costs. This section characterizes these costs formally and through simulation, and considers which constructions are likely to perform well in real environments.

6.1. Analysis

Each OA construction makes tradeoffs on the consumption of resources (e.g., storage versus computational costs). This section and Table 2 describe the computational and storage costs associated with the origin authentication constructions. The following notation is used throughout. The number of delegations made by ownership source is m, and the number of delegations made to a particular AS or organization is j. The verifier is validating n proofs associated with k unique ASes and organizations. We denote the constant (size) quantity ϕ as signature size, α as AS/organization identifier size, and μ as the output size of the hash function used by the tree constructions.

In simple delegation attestations, the verifier acquires a signed statement (proof). Verification requires a signature validation per assertion, and the storage costs are the sum of the size of the proofs. In the authenticated delegation list and the AS authenticated delegation list, the verifier acquires a signed list of either the entire list of delegations or the delegations associated with a particular AS or organization, respectively. Hence, the verifier will perform either 1 or k signature operations to validate the prefixes. The storage costs are one signature plus the number of prefixes, or k signatures plus the number of prefixes associated with those ASes/organizations.

The verifier need only validate a single signature in all tree schemes. This represents a minimal cost, and can be used to vastly reduce the computational requirements placed on verifiers. The storage costs associated with authentication delegation trees are dependent on the locality of reference. That is, the costs are low where the proofs have common ancestors in the proof tree.

The storage costs of each approach are /24 illustrated through the following fictional example. Assume that a signature size is 110 bytes (from [29], $\phi = 110$), four-byte AS/organization identifiers ($\alpha = 4$), and the output of the hash function is 16 bytes (e.g., as per MD5 [52], $\mu = 16$), and that the verifier is validating 100 prefixes (out of 1000 issued by an ownership source, $n = 100$, $m = 1000$) associated with 20 unique ASes/organizations (evenly, $k = 20$, $j = 50$). The space used is 11400 bytes for simple attestations, 4110 bytes for authenticated delegation lists, 6200 bytes for AS authenticated delegation lists, and 2110–8510 bytes for an authentication delegation tree.

Table 2
Resource usage—the number of signature and hash operations, and storage costs of each origin authentication construction at a verifying BGP speaker

Construction	Sig.	Hash	Appx. Storage
Simp Del Attest	n	n/a	$n(\phi + \alpha)$
Auth Del List	1	n/a	$m\alpha + \phi$
AS Auth Del List	k	n/a	$k(\phi + j\alpha)$
Auth Del Tree *min*	1	n	$\phi + n(\mu + \alpha)$
Auth Del Tree *max*	1	$n \log \frac{m}{n}$	$\phi + n\mu \log \frac{m}{n} + n\alpha$

6.2. Simulation

It is not immediately clear which of the several origin authentication service designs is the most appropriate for the Internet. In this section, we evaluate origin authentication services via trace-based simulation. For simplicity, we do not simulate authentication dictionaries. Obtained from the RouteViews corpus, all experiments use a trace of BGP updates arriving at a single BGP speaker on April 2, 2003. The trace contains 653,649 UPDATE messages recorded over a 24-h period (midnight to midnight). The delegation graph used in the following simulations was derived using the algorithm defined in Section 5.2 A Routeviews RIB taken at midnight on April 3, 2004—the router table occurring at the end of the test data– was used as input for the algorithm. The delegation graph defines all prefix ownership and the associated delegation chains, and hence is crucial in determining to the performance of any approach.

The *OAsim* simulator models the operation of a single BGP speaker. After preprocessing a delegation map, this simulator accepts timed BGP UPDATE streams and computes the costs associated with the validation and storage of the associated origin authentication proofs. OAsim implements four service designs modeled in the previous section: *simple attestations*, *authenticated delegation lists*, *AS authenticated delegation lists*, and *authentication delegation trees*. The simulator maintains a variable size (LRU) cache which models the unique storage costs of each approach. Proof sizes are derived using the formulas presented in the previous section. We assume that all certificates are stored locally (e.g., not considered when calculating cache sizes).

In all tests, we model online operation as transmitting delegation and assignment proofs through the BGP optional transitive attributes [55]. The

bandwidth experiments ignore the current BGP MTU (4096 bytes). We seek to understand the efficacy of optimal solutions, and as such, relax this systemic limitation. Note that the only construction likely to be frequently affected by the MTU limitation is the authenticated delegation list. The modeled off-line schemes simply acquire proofs from external entities where cached values do not provide sufficient validation (e.g., S-BGP repositories, IRVs).

A first battery of tests makes a broad comparison of the origin authentication methods. Fig. 6 shows the computational costs as measured by signatures in 5-min increments of the 24-h trace period (for legibility, the figures only show a representative 4-h period during the trace). In all schemes, signature validation dominates other computational costs (e.g., parsing, hashing, etc.), and hence, is a good estimate of overall computation. The most costly solution is the simple attestation; this stands to reason as every (uncached) UPDATE leads to a signature validation. This is followed by the AS authenticated delegation lists which incur a half to a third fewer signatures.

The authenticated delegation lists and authentication delegation trees are more efficient—both require at times an order of magnitude less computation than simple attestations. Delegating organizations in these schemes issue proofs for all delegations simultaneously. Hence, a large cache (in this case 1M) eliminates the need for many validations. The authentication delegation trees are generally more effective because each authentication delegation tree proof is cached separately.

A second set of tests compare the costs of on-line and off-line OA. As depicted in Fig. 7, bandwidth costs in online OA are discrete. Authenticated delegation lists are significantly more expensive than the other schemes because each UPDATE must be accompanied with a complete proof. Most delegations are made by one of a few entities, and hence, are part of naturally large proofs. All other proofs are of a relatively constant size, which are small with respect to authenticated delegation lists.

Not shown, off-line bandwidth costs are nominal. In no period was more than 100k of bandwidth consumed for any construction, and in most periods, less than 10k were consumed. This stands to reason, as very few proofs (10s) are validated in any period. The only exception to this was a spike of several hundred kilobytes of data associated with simple proofs and the authenticated tree scheme. This spike was a result of a large block of deaggregated addresses. As a result, the verifier had to continually acquire (but not verify) many proofs.

A third set of tests sought to evaluate the degree to which caching can improve performance. The delegation graph defined in the preceding section contained 114,183 delegations at the time of this simulation. Caching all proofs for these delegations requires 13.4M cache for simple attestations, 1.2M for authenticated delegation lists, 4.0M for AS

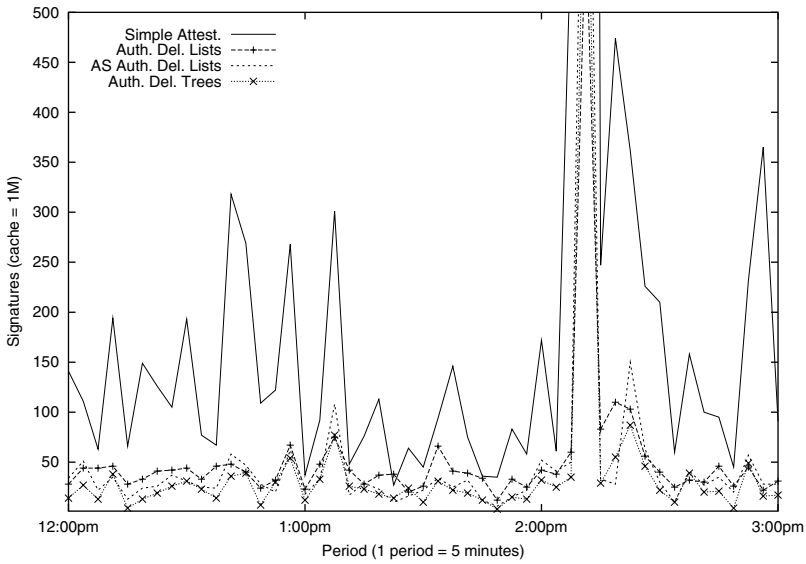

Fig. 6. Computational cost—signature validations of each origin authentication scheme. Experiments performed with a 1MB warm cache.

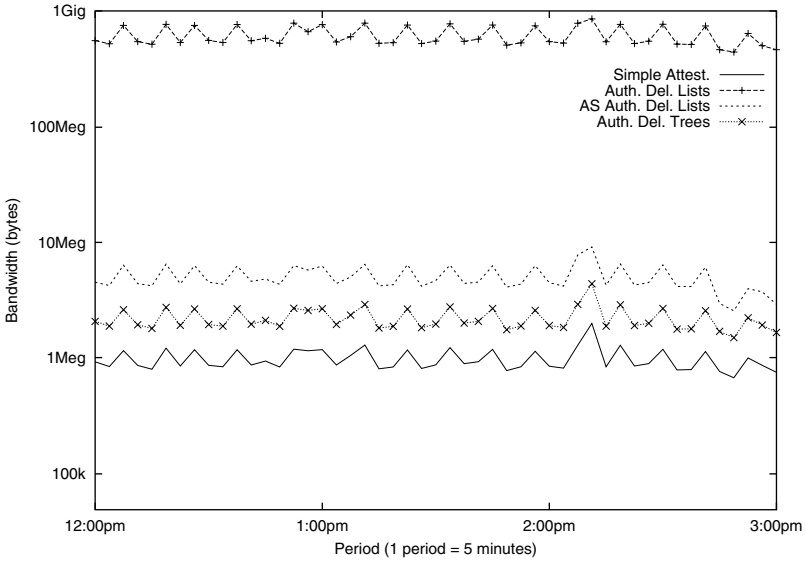

Fig. 7. Bandwidth cost—bandwidth costs of origin authentication schemes. Experiments performed with a 1MB warm cache.

authenticated delegation lists, and 4.7M for authentication delegation trees. Note that these figures support the previous analysis, where authenticated lists, due to the large number of proofs validated by a single organizational signature, forms the minimal storage requirement. Conversely, simple attestations, where a signature is required for every proof, forms the maximal storage limitation. AS authenticated lists have a signature per AS, which necessarily requires more storage than the authenticated list (due to organizations owning multiple ASes) and trees are variable in size, depending on reference locality. In this case, the tree scheme requires slightly more storage than the AS authenticated list.

Figs. 8 and 9 show the computational costs associated with each scheme under varying cache sizes over a 3-h period (4:40 p.m.–9 p.m.). The 100-megabyte cache far exceeds the size of the proofs, and hence measures only new proofs (the test starts with a cold cache the preceding midnight). Medium-sized cache sizes (1M and 100k) are affected by reference locality. The most notable aspect of these graphs is the degree to which the tree scheme outperforms the others. This is due to two factors: the structure of the delegation graph and the use of succinct proofs. Because 16 proofs encompass 80% of the delegations, the associated signatures are likely to be present in the cache. Because of their size, the succinct proofs are likely to remain in the cache.

After removing the load associated with organization-to-AS delegation (the leaf delegation in the graph), authenticated delegation lists were shown to outperform AS authenticated delegation lists. This is again due to delegation density: an AS is likely to see many delegations from a single organization within some temporal bounds, regardless of to whom they are delegated. More generally, this demonstrates that delegator-centric solutions are well suited to current BGP UPDATE traffic.

These results lead to a new cache strategy for aggregate proof schemes: *caching organization-to-organization delegation proof signatures only*, rather than the entire proof structures. A complete cache of these signatures would be just over 200 kilobytes. Because verification would perform as if all proofs were previously cached, the computational cost could be significantly reduced for authenticated lists. This would mitigate the thrashing effect of large proof approaches on small caches (lists). However, this solution would offer little added benefit to tree-based solutions, as they already offer caching that is close to optimal. Additionally for trees, because the structure itself is not cached, the intermediate hashes comprising the Merkle hash tree represented by the tree structure would need to be calculated when the structure is received. Because of the low cost of calculating hashes compared to signature validations, though—with a hash operation requiring three orders of magnitude less computation time [20]—the additional costs of this

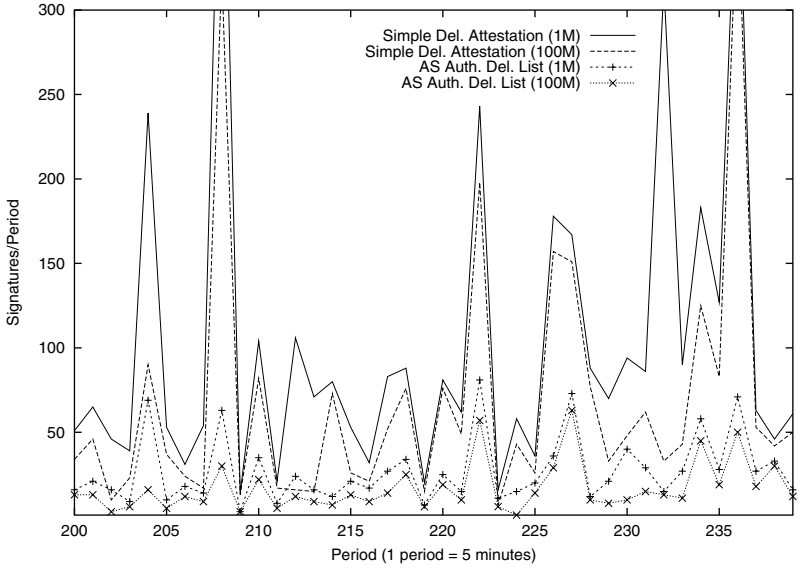

Fig. 8. Cache evaluation—signature validations for attestations and AS authenticated delegation lists.

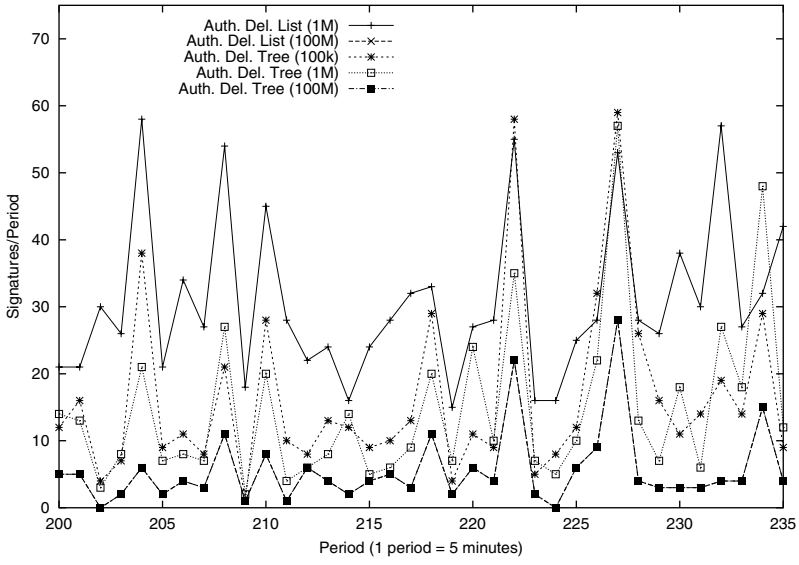

Fig. 9. Cache evaluation—signature validations under authenticated delegation lists and trees.

scheme are minimal. Importantly, no changes are necessary to implement this strategy from a proof structure standpoint, and there are no changes in the formalism; only the system implementation itself would change.

We will consider these and other strategies (e.g., LFU caching disciplines) in future work. In summary, of the schemes that have been considered, delegation trees provide the most balanced overall solution due to their low computational and bandwidth requirements and average storage overhead.

The assertions we make about the feasibility of in-band origin authentication using the cryptographic constructions presented were experimentally evaluated in [60]. Through large-scale discrete-event simulation of a 110-AS network, it was shown that efficiencies of our authentication constructions, particularly the list and tree approaches, greatly reduced the number of required signature validations, and the convergence time recorded was such that in-band delegation is possible.

6.3. Approximation sensitivity

Assume that our approximation of the delegation graph is completely wrong: the IP address delegation graph changes frequently and contains no nodes of high degree (e.g., low delegation density). This would indicate that address space ownership is highly fluid and fragmented. This is counter to almost all studies of BGP, and would signal larger problems with interdomain routing. Such features, if true, would markedly increase the size of BGP tables, increase the BGP load, and prevent timely convergence (e.g., in the limit aggregation becomes useless). This does not seem likely.

Now assume the more likely event that we have underestimated the number of ownership sources and delegations in the Internet. This is almost certainly true—we have worked from incomplete information about organizational delegation. We argue this is a reflection of the BGP data itself: providers and registries hand out blocks to organizations, not ASes. However, operational evidence strongly suggests that it is infrequently the case that the address space is further delegated. Hence, we claim that the approximation is of a high enough quality to draw general conclusions.

The effect of a larger body of ownership sources and number of delegations will affect our results quantitatively but not qualitatively. Lesser delegation densities close the performance gap between the different designs. Similarly, a larger number of delegations will only serve to scale up resource costs on all schemes. In both cases, the wide gulf between measured costs signals that even a gross approximation is sufficient to characterize the constructions.

7. Conclusions

The lack of security in interdomain routing protocols is increasingly recognized as an urgent problem. An important aspect of any comprehensive approach is the means by which it performs *origin authentication*. An origin authentication service traces and validates the delegation of address usage from authorities to organizations, and ultimately to the ASes that originate them. Previous works have identified simple solutions, but no work has defined and generalized origin authentication or evaluated solutions using a complete picture of delegation on the Internet.

This paper has developed a broad understanding of the issues, designs, and practicality of origin authentication services. This work is composed of three serial efforts: formalization, modeling, and simulation. We initially formalized the semantics of address advertisements and proofs of delegation. Broad classes of origin authentication services are defined by creating novel cryptographic proof systems. We classify the current delegation of IPv4 address space by modeling the *address delegation graph* from current interdomain routing data and public registry information. An analysis of this graph shows that the current delegation on the Internet is largely static and dense: 16 entities perform 80% of the address delegation. The *OAsim* simulator uses our approximate delegation graph and BGP announcements to compute the resource consumption of origin authentication services. Our simulation experiments show that resource costs can be significantly reduced by using proof systems centered on the delegator organizations and ASes. Experiments with these systems show that resource costs can be reduced by an order of magnitude over current proposed solutions. Such results indicate that on-line origin authentication may now be within the realm of possibility.

Securing the current interdomain routing infrastructure is likely to be a lengthy process. The security and networking communities must continually reevaluate the assumptions and environments upon which the solutions are based. Work such as this serve as important contributions to this process; a thorough understanding of the trade-offs inherent to these services is essential. As a chief motivation of this work, such understanding must be grounded in current realities of the Internet. It is only through the cumulative force of this and similar works that breakthroughs in interdomain routing security can be made.

References

[1] S. Agarwal, C.-N. Chuah, S. Bhattacharyya, C. Diot. The impact of BGP dynamics on intra-domain traffic, New York, NY, USA, June 2004, ACM SIGMETRICS 2004.

[2] W. Aiello, K. Butler, P. McDaniel, Path authentication in interdomain routing, Technical Report NAS-TR-0002-2004, Network and Security Research Center, Department of Computer Science, Pennsylvania State University, University Park PA, USA, October 2004.

[3] W. Aiello, F. Chung, L. Lu, Random evolution of massive graphs, in: Proceedings of IEEE Symposium on Foundations in Computer Science, IEEE, Las Vegas, NV, 2001, pp. 510–519.

[4] W. Aiello, S. Lodha, R. Ostrovsky, Fast digital identity revocation, in: Proceedings of CRYPTO 98, Santa Barbara, CA, August 1998, pp. 137–152.
[5] ARIN, American Registry for Internet Numbers, March 2005. Available from: <http://www.arin.net/>.
[6] A. Barabási, R. Albert, Emergence of Scaling in Random Networks, Science 286 (1999) 509–512.
[7] S. Bellovin, R. Bush, T. Griffin, J. Rexford, Slowing routing table growth by filtering based on address allocation policies, June 2001. Available from: <http://www.research.att.com/jrex/>.
[8] K. Butler, T. Farley, P. McDaniel, J. Rexford, A survey of BGP security issues and solutions, Technical Report TD-5UGJ33, AT&T Labs-Research, Florham Park, NJ, February 2004 (revised June 2004).
[9] S. Cheung, An efficient message authentication scheme for link state routing, in: 13th Annual Computer Security Applications Conference, San Diego, CA, December 1997, pp. 90–98.
[10] B. Weis (Ed.), Secure origin BGP (soBGP) certificates, Internet Research Task Force, June 2003 (draft-weis-sobgp-certificates-00.txt).
[11] R. White (Ed.), Deployment considerations for secure origin BGP (soBGP), Internet Research Task Force, October 2002 (draft-white-sobgp-bgp-extensions-00.txt).
[12] M. Faloutsos, P. Faloutsos, C. Faloutsos. On power-law relationships of the Internet Topology, in: Proceedings of ACM SIGCOMM Conference, ACM, Cambridge, MA, 1999.
[13] N. Feamser, H. Balakrishnan, Detecting BGP configuration faults with static analysis, in: Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI'05), USENIX, Boston, MA, May 2005.
[14] V. Fuller, T. Li, J. Yu, K. Varadhan, Classless inter-domain routing (CIDR): an address assignment and aggregation strategy, Internet Engineering Task Force, RFC 1519, September 1993.
[15] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, A. Rubin, Working around BGP: an incremental approach to improving security and accuracy of interdomain routing, in: Proceedings of Network and Distributed Systems Security 2003, Internet Society, San Diego, CA (Draft), February 2003.
[16] B. Green, BGP security update: is the sky falling? NANOG 25, June 2002.
[17] T. Griffin, Personal communication, May 2003.
[18] R. Housley, W. Ford, W. Polk, D. Solo, Internet X.509 Public Key Infrastructure Certificate and CRL Profile, Internet Engineering Task Force, RFC 1949, January 1999.
[19] Y. Hu, A. Perrig, D. Johnson, Efficient security mechanisms for routing protocols, in: Proceedings of Network and Distributed Systems Security 2003, Internet Society, San Diego, CA, February 2003.
[20] Y.-C. Hu, A. Perrig, M. Sirbu, SPV: secure path vector routing for securing BGP, in: Proceedings of ACM SIGCOMM'04, Portland, OR, USA, ACM, August 2004.
[21] G. Huston, Bgp table data, March 2005. Available from: <http://bgp.potaroo.net/>.
[22] IANA, Autonomous System Numbers, March 2005.
[23] IANA, Internet Protocol V4 Address Space, March 2005. Available from: <http://www.iana.org/assignments/ipv4-address-space>.
[24] IANA, The Internet Assigned Numbers Authority, March 2005. Available from: <http://www.iana.org/>.
[25] ICANN, The Internet Corporation for Assigned Names and Numbers, May 2003. Available from: <http://www.icann.org/>.
[26] S. Kent, Securing the Border Gateway Protocol: a status update, in: Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, Torino, Italy, October 2003.
[27] S. Kent, R. Atkinson, Security architecture for the Internet Protocol, RFC 2401, November 1998.
[28] S. Kent, C. Lynn, J. Mikkelson, K. Seo, Secure Border Gateway Protocol (S-BGP)—real world performance and deployment issues, in: Proceedings of Network and Distributed Systems Security 2000, Internet Society, February 2000.
[29] S. Kent, C. Lynn, K. Seo, Secure Border Gateway Protocol (Secure-BGP), IEEE Journal on Selected Areas in Communications 18 (4) (2000) 582–592.
[30] P. Kocher, On certificate revocation and validation, in: R. Hirschfeld (Ed.), Financial Cryptography, vol. 1465, Springer, Anguilla, British West Indies, 1998, pp. 172–177, February.
[31] C. Kruegel, D. Mutz, W. Robertson, F. Valeur, Topology-based detection of anomalous BGP messages, in: Proceedings of the 6th Symposium on Recent Advances in Intrusion Detection (RAID), September 2003, pp. 17–35.
[32] C. Lonvick, RADIUS attributes for soBGP support, Internet Draft, February 2004.
[33] R. Mahajan, D. Wetherall, T. Anderson, Understanding BGP misconfiguration, in: Proceedings of ACM SIGCOMM '02, ACM, September 2002.
[34] P. McDaniel, S. Jamin, Windowed certificate revocation, in: Proceedings of IEEE INFOCOM 2000, IEEE, Tel Aviv, Israel, March 2000, pp. 1406–1414.
[35] X. Meng, Z. Xu, L. Zhang, S. Lu, An analysis of BGP routing table evolution, Technical Report TR030046, Computer Science Department, UCLA, January 2003.
[36] R. Merkle, Protocols for public key cryptosystems, in: Proceedings of the 1980 Symposium on Security and Privacy, IEEE, Oakland, CA, April 1980, pp. 122–133.
[37] D. Meyer, The RouteViews Project, May 2003. Available from: <http://www.routeviews.org/>.
[38] S. Micali, Efficient certificate revocation, Technical Report Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, 1996.
[39] S. Misel, Wow, AS7007! Available from: <http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html>.
[40] S. Murphy, BGP security vulnerabilities analysis (Draft), Internet Research Task Force, February 2002 (draft-murphy-bgp-vuln-00.txt).
[41] M. Naor, K. Nassim, Certificate revocation and certificate update, in: Proceedings of the 7th USENIX Security Symposium, January 1998, pp. 217–228.
[42] H. Narayan, R. Govindan, G. Varghese, The impact of address allocation and routing on the structure and implementation of routing tables, in: Proceedings of

ACM SIGCOMM'03, Karlsruhe, Germany, ACM, August 2003.
[43] J. Ng, Extensions to BGP to support secure origin BGP (soBGP), Internet Draft, October 2002.
[44] D. Nicol, S. Smith, M. Zhao, Efficient security for BGP route announcements, Dartmouth Computer Science Technical Report TR-2003-440, February 2002.
[45] O. Nordström, C. Dovrolis, Beware of BGP attacks, Computer Communications Review 34 (2) (2004) 1–8.
[46] D. Pei, W. Aiello, A. Gilbert, P. McDaniel, Origin disturbances in BGP, July 2004.
[47] D. Pei, D. Massey, L. Zhang, A framework for resilient Internet routing protocols, Technical report, UCLA, November 2003.
[48] R. Perlman, Network layer protocols with Byzantine robustness, Technical Report MIT/LCS/TR-429, October 1988.
[49] Y. Rekhter, T. Li, A Border Gateway Protocol 4 (BGP 4), Internet Engineering Task Force, March 1995, RFC 1771.
[50] J. Rexford, Personal communication, April 2005.
[51] J. Rexford, J. Wang, Z. Xiao, Y. Zhang, BGP routing stability of popular destinations, in: ACM SIGCOMM IMW (Internet Measurement Workshop) 2002, 2002.
[52] R. Rivest, The MD5 Message Digest Algorithm, Internet Engineering Task Force, RFC 1321, April 1992.
[53] K. Seo, C. Lynn, S. Kent, Public-key Infrastructure for the Secure Border Gateway Protocol (S-BGP), in: Proceedings of DARPA Information Survivability Conference and Exposition II, IEEE, June 2001.
[54] B. Smith, J. Garcia-Luna-Aceves, Securing the border gateway routing protocol, in: Proceedings of Global Internet '96, November 1996, pp. 103–116.
[55] J. Stewart, BGP4: Interdomain Routing in the Internet, Addison-Wesley, 1998.
[56] L. Subramanian, S. Agarwal, J. Rexford, R.H. Katz, Characterizing the Internet Hierarchy from multiple vantage points, in: Proceedings of IEEE INFOCOM 2002, IEEE, June 2002.
[57] L. Subramanian, V. Roth, I. Stoica, S. Shenker, R. Katz, Listen and whisper: security mechanisms for BGP, in: Proceedings of the First Symposium on Networked Systems Design and Implementation (NSDI'04), San Francisco, CA, USA, March 2004.
[58] T. Wan, E. Kranakis, P.C. van Oorschot, Pretty secure BGP (psBGP), in: Proceedings of Network and Distributed Systems Security Symposium 2005 (NDSS'05), San Diego, CA, USA, Internet Society, February 2005.
[59] Z. Wenzel, J. Klensin, R. Bush, S. Huter, Guide to administrative procedures for the Internet infrastructure, Internet Engineering Task Force, August 2000, RFC 2901.
[60] M. Zhao, S.W. Smith, D.M. Nicol, Evaluating the performance impact of PKI on BGP security, in: Proceedings of the 4th Annual PKI Research Workshop (PKI'05), NIST, Gaithersburg, MD, April 2005.
[61] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S.F. Wu, L. Zhang, An analysis of BGP multiple origin AS (MOAS) conflicts, in: ACM SIGCOMM Internet Measurement Workshop 2001, ACM, November 2001.
[62] G.K. Zipf, Human Behaviour and the Principle of Least Effort, Hafner, 1949.

Patrick McDaniel is the Hartz Family Career Development Professor in the Computer Science and Engineering Department at the Pennsylvania State University. He received his Ph.D. from the University of Michigan in 2001 where he studied the form, algorithmic limits, and enforcement of security policy. Prior to joining Penn State, Patrick was a senior technical staff Member of the Secure Systems Group at AT&T Labs-Research and Adjunct Professor of the Stern School of Business at New York University.

His recent research efforts have focused on security management in distributed systems, network security, and public policy and technical issues in digital media. Patrick is a past recipient of the NASA Kennedy Space Center fellowship, a frequent contributor to the IETF security standards, and has authored many papers and book chapters in various areas of systems security. He is currently serving as the Program Chair of the 2005 USENIX Security Symposium, the Vice Chair for Security and Privacy for WWW 2005, and is the Chair of the Industry and Government Track at the ACM Computer and Communications Security conference. Patrick is also an associate editor of the journal ACM Transactions on Internet Technologies. Prior to pursuing his Ph.D. in 1996, Patrick was a software architect and program manager in the telecommunications industry.

William Aiello is the Department Head of the UBC Department of Computer Science. Previously, he was at AT&T Research Labs in New Jersey where he was Division Manager of the Network Security and Cryptography Research Group. In his role there he managed eight researchers in the fields of network security, cryptography, and data privacy; development and guidance of a diverse research program in network security with a mission to provide both significant internal impact on AT&Tas networks and services, as well as external technical leadership through contributions to standards, conferences and journals. He holds a Ph.D. from MIT in Applied Mathematics with research in Cryptography and Complexity Theory under the direction of Prof. Shafi Goldwasser and prior to that graduated from Stanford University with Distinction in Physics.

Kevin Butler is a Ph.D. Candidate in the Systems and Internet Infrastructure Security Laboratory in the Department of Computer Science and Engineering at the Pennsylvania State University. He received his MS in electrical engineering at Columbia University in 2003, and his B.Sc. in electrical engineering at Queen's University at Kingston, Ontario in 1999.

He worked extensively with BGP and wide-area networking at UUNET, and as a research scientist in the Applied Research group at Telcordia Technologies (formerly Bellcore) he examined multimedia over IP

and network operations issues. He completed an extended internship in the Secure Systems Group at AT&T Labs-Research, working with Dr. Patrick McDaniel, now his advisor at Penn State. His research interests focus on security in large-scale systems, particularly cryptographic solutions derived through data analysis.

John Ioannidis (“JI”) is a Senior Research Scientist at Columbia University. The underlying theme of his research has been protecting large-scale infrastructures. In recent years, he was worked on ways of improving the state of interdomain routing with emphasis on scalable and incrementally deployable protocols. He has also worked on methods to defend against distributed denial of service attacks, with emphasis on solutions that are incrementally deployable. He has also done extensive work on Trust Management and its applications. Older work includes the original Mobile-IP work, swipe (the precursor to IPsec), and the original implementations of IPsec for BSD and FreeS/WAN.