Origin Authentication in Interdomain Routing

William Aiello
AT&T Labs - Research
Florham Park, NJ
aiello@research.att.com

John Ioannidis
AT&T Labs - Research
Florham Park, NJ
ji@research.att.com

Patrick McDaniel
AT&T Labs - Research
Florham Park, NJ
pdmcdan@research.att.com

General Terms

Security

Keywords

routing, security, address management, BGP, delegation

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection

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 enables not only attacks by malicious entities, but indirectly allow seemingly inconsequential miconfigurations 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 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 traced based simulation. This simulation shows the enhanced proof systems can reduce resource consumption by an order of magnitude over currently proposed solutions.

1. INTRODUCTION

Routing in 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 result in 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. Intradomain and interdomain routing decisions are largely made independently.

The Border Gateway Protocol [25, 30] is the interdomain routing protocol used on the Internet. BGP routing domains, called *Autonomous Systems* (ASes) announce IP address ranges, called *prefixes* to its 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 [29, 16, 22, 8], and interim and long-term solutions are seeking broad adoption [15, 8, 6]. 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 [13]), 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 them the right to to do so.

The lack of authenticated origin information is increasingly viewed as a critical vulnerability of the Internet infrastructure [9]. 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 and led to widespread outages [21]. Similarly, Zhao *et al.* found that there are a great many causes that multiple ASes claim to be the origin of a single prefix (called a MOAS conflict), almost all of them bad [33]. The authors found that *prefix hijacking* due to apparent misconfiguration was a frequent cause of MOAS conflicts. Other outages were similarly enabled by lack of validation of origin and routing information [17].

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 ASes. We show that the semantics of address delegation mandates that any path (i.e., delegation chain)

CCS'03, October 27–30, 2003, Washington, DC, USA.
Copyright 2003 ACM 1-58113-738-9/03/0010 ...$5.00.

[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).

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 is 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 authority over a specified address range, b) that an AS has been granted the right to be the origin of that address range, or c) that the address range cannot be used (reserved). Verifiers collect and validate proofs corresponding to the delegation chains. We apply a range of cryptographic constructions 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 vast majority of delegation is performed by a few entities: 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 traced 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 uncovered 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 that consolidate proofs by delegator can significantly reduce resource costs.

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 [16], IRV [8]).

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] [13]. IANA *delegates* parts of the global address space to organizations representing commercial, public, or other interests [32]. 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 (AS) 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) may maintain more than one. Each organization *assigns* 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. Figure 1 illustrates several common ways that address space is delegated to organizations and assigned to ASes.

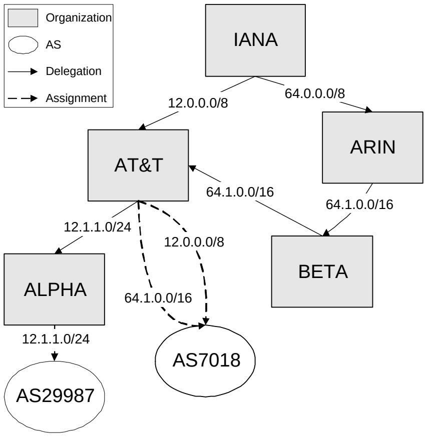


Figure 1: IPv4 address management - All ownership of IPv4 address is *delegated* by IANA to organizations which may delegate further. Addresses are *assigned* to an AS for advertisement via BGP.

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 discovered that having a single body governing all delegation was administratively difficult. Hence, registries like ARIN [3] were introduced to delegate address space received from 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&Ts AS. In this case, there is no need for delegation because DELTA's address space is total encompassed by AT&T (both in the logical and physical sense). Now consider an another organization ALPHA that is also a customer of AT&T but wishes to run its own AS. 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 which 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 12.1.1.0/24 and 64.1.0.0/16), as does ALPHA (AS2997 is assigned 12.1.1.0/24).

AT&T retains control (originates) of 12.0.0.0/8 by assigning the prefix to AS7018. This assignment is seemingly ambigu-

[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.

ous: 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&T's delegation and ALPHA's 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 as well as 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 [28]. 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 [29, 5, 22, 24]. The identified problems are succinctly described by Murphy in her analysis of BGP [22]. Her analysis shows that the vulnerabilities of BGP directly flow from the following truths: *a*) messages do not have guaranteed integrity, freshness, or authenticity, *b*) paths are not authenticated, and *c*) there is no way to validate an AS's authority to advertise a prefix. This paper focuses solely on this last point, the lack of authenticated address usage. As identified by Murphy and others, 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) [28]. 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 [16, 15]. Origin authentication is supported in S-BGP by an address allocation public key infrastructure (PKI). Authorities in the S-BGP PKI issue certificates mapping 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 signed statements that indicate an AS has the right to advertise a prefix (i.e., delegates to the AS). Others have applied more complex, but often efficient, cryptographic structures to the problem of path-vector security [10].

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 validation can be completed by repositories. Centralized attestation repositories mitigate the costs of validation during *table resets* (e.g., memory re-initialization following a router reboot). For example, router can rely on the repository to assert validity, rather than by validating received or acquired proofs.

One challenge in the adoption of any interdomain routing security solution is its integration with existing infrastructure. In the Interdomain Routing Validation (IRV) project [8], 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, application) 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 requestor for optimization or access control.

The emerging soBGP protocol combines proactive security measures with anomaly detection [6]. Like IRV, the proposed soBGP protocol focuses on incremental deployment. soBGP validates address announcements in a way similar 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. Because no structure of authorities is imposed, communities of soBGP ASes can quickly bootstrap and grow independently.

Whether by constructing and distributing cryptographic proofs or by detecting divergence from received policy data, the works described above largely share a single goal: the verification of address delegation and its subsequent use. Hence, inasmuch as this paper provides important general insights and solutions to this problem, we strongly argue that any approach must be informed by this or similar works.

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 by precisely defining the relations that will be authenticated.

Definitions: 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 $\mathcal{ASN} = \{1, 2, \ldots, K\}$ be the set of all Autonomous System Numbers, where currently $K = 2^{16}$. Let $\mathcal{S}$ be the set of all BGP speaking organizations, i.e., those organizations to which AS numbers have been assigned by ICANN [14]. For each organization $C \in \mathcal{S}$, let $\mathcal{ASN}(C)$ be the set of AS numbers currently assigned to it. Let $\mathcal{O}$ be all of the organizations in $\mathcal{S}$ plus IANA and the other prefix registries. $\mathcal{O}$ is the set of all organizations which 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 $\mathcal{IPA} = \{0, 1\}^\ell$ be the set of all ℓ-bit IP addresses where $\ell = 32$ for IPv4 and $\ell = 64$ 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 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, $x/j = \{x \cdot y \,|\, y \in \{0,\, 1\}^{\ell - j}\}$ which is simply all of the ℓ-bit addresses with the j most significant bits equal to x. (By convention, $\{0,\, 1\}^0 = \emptyset$ so that $\emptyset/0 = \mathcal{IPA}$ is the set of all addresses.) Using this notation x/j is equal to the disjoint union of $x \cdot 0/(j+1)$ and $x \cdot 1/(j+1)$. 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 $\emptyset/0 = \mathcal{IPA}$, 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.)

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. More general options are discussed subsequently.

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

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, $C \neq \mathrm{IANA}$ 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 $V = \mathcal{O} \cup \mathcal{ASN} \cup \{\mathrm{R}\} \cup \{\bot\}$. 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 $\bot$. 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 $\mathcal{O} \cup \mathcal{ASN} \cup \{\mathrm{R}\} \cup \{\bot\}$. Thus, no node in $\mathcal{O}$ is a terminal.

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

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.

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.

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. But the local connectivity of C in the delegation graph is not enough information to rule out being in a cycle when organizations which 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.

Null Assignments: As defined, a valid path for y/k may have an assignment edge from C to $\bot$ 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 the all the prefixes that are subsets of their $x/8$, except for those they have further delegated. A provider's policy determines which of the sub prefixes it will pair with which of its AS numbers in BGP UPDATE announcements and which sub prefixes it decides not to announce, at least until it's policy changes. In practice, only a small fraction of the possible sub prefixes actually appear in announcements (we establish see Section 5).

Uniqueness: The definitions thus far do not rule out the following: a delegation graph that is a directed tree rooted at IANA where every path is valid. To see this consider the case in which a valid partial delegation path ends in C, and suppose that C has even received

[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.

[6] We adopt the term *attestation* from Kent et. al. [16]. In the vernacular, attestations are proclamations of truth, and serve as good metaphors for statements of address delegation.

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.

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 [28], 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.

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 (argument omitted due to lack of space). 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 $\mathcal{O}$ or be the second to last node in $\mathcal{O}$.

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 restricted the number of valid delegations 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, are unique.

4.1 Origin Authentication Tags and Delegation Attestations

In our scheme origin announcements are verified by *Origin Authentication Tags*, or OATs. OATs 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 is outside the scope of this paper. See the description of S-BGP PKI [28] 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 $\mathcal{D}(C)$ 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, in our scheme all parties 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 $x/j \in \mathcal{D}(C)$ is delegated to some organization C' then no prefix that is a strict subset of x/j is in $\mathcal{D}(C)$.

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 sub prefixes by another organization. And 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 sub prefixes 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 sub prefix 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 sub prefixes 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 get routed to AS n_1 rather than AS n_2. Such attacks are inherent to the longest prefix match heuristic.

4.2 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.

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

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

$$\begin{array}{l} [(x_1/j_1,\ F_C(x_1/j_1))]_C\,, \\ [(x_2/j_2,\ F_C(x_2/j_2))]_C\,, \\ \ldots, \\ [(x_s/j_s,\ F_C(x_s/j_s))]_C \end{array}$$

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

Consider an example of an OAT for the origin announcement $(12.1.1.0/24,\ \mathrm{AS29987})$ from Figure 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

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

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.

In practice, simple attestations are signed statements binding the prefix to a organization identifier. It is incumbent on the assumed certificate management infrastructure to issue and manage the identifiers. Note that unlike the design of S-BGP [28] 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 [28]. 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 burden on the both the issuing organization and the verifiers' (routers) [15] (see Section 6 for further analysis).

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

$$\begin{array}{lcl} [& (x_1/j_1,\ F_C(x_1/j_1)), & \\ & (x_2/j_2,\ F_C(x_2/j_2)), & \\ & \ldots, & \\ & (x_s/j_s,\ F_C(x_s/j_s)) &]_C \end{array}$$

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

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, some organizations' authenticated delegation lists 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 are evaluated experimentally below and compared to 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 [16]. The advantage of this approach is 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.

Authenticated Delegation Tree Consider the following scheme. An organization C creates a Merkle hash tree [18]. The values of the leaves of the tree are of the form $(x/j,\ F_C(x/j))$ for each $x/j \in \mathcal{D}(C)$. 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 [10]).

In this scheme, the delegation attestation that C is delegating/assigning x/j to $F(x/j)$ consists of the value 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 then verify C's signature on h_0. The size of a single proof is logarithmic in the size of $\mathcal{D}(C)$. 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.

Authenticated Delegation Dictionaries Naor and Nissim introduced the notion of authenticated dictionaries [23] 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 [23] 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 $y/k \in \mathcal{D}(C)$. To make this concrete we briefly describe the scheme in [23] modified to this context.

We start with a search trees in which the leaves are sorted, say, left to right, based on the search key. For the sake of efficiency [23] 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 lexicographically 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 $j \geq 0$ and $k \geq 0$ respecting $0 \leq j+k \leq \ell$, 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 smalles elment 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 $y/k \in \mathcal{D}(C)$, and they are 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 the 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 than 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 sub prefixes of y/k should be empty. That is, none of the proper sub prefixes 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 subprefixed 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 [28]. 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. 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.

4.3 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 prefixed to a customer, or the customer owns an address prefix and delegates it to the provider. See Figure 1.) In these cases the expiration in the delegation attestation would naturally be set to the expiration date of the customer/provider service agreement.

BGP is a delta-based protocol in that routing information is propagated reliably only as changes in the network occur. Consider the case where an origin announcement is propagated on day 1 and some delegation attestation in the prefix delegation path is set to expire at the end of day 2. Given that BGP is a delta-based protocol, what is the status of the route for that prefix on day 3? Due to space limitations we defer a complete discuss of these issues.

Replay protection can easily be achieved if delegation attestations are retrieve out-of-band by verifiers over a secure channel (e.g., TLS) from a directory. In-band delivery of delegation attestation 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 [20, 2] 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. Again, we omit a full discussion of these issues due to space limitations.

4.4 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 due go through n can be aggregated by AS n who can send out an announcement for $(y/j, n)$ with a valid OAT. A slightly larger set of aggregation alternatives for C are possible using the generalizations to our scheme discussed in the appendix.

5. THE ADDRESS DELEGATION GRAPH

The cost of origin authentication systems in general, and the constructions defined in the preceding sections in specific, 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.

5.1 Approximating IP Address Delegation

While previous studies have accurately reconstructed the routing topology graph [31], it is exceptionally difficult to approximate a delegation graph for the Internet. To show why this is so, consider the fragmentation of AT&T's `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. 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 be intractable.

In a related work, Kent et.al. estimated the statistical properties of the IPv4 address delegation in investigating deployment costs of S-BGP [15]. 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 which 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 (1,500 in 1999 vs. 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 hence we focus on the largest visible load on any BGP speaker. The ratio of iBGP to eBGP traffic is topology and AS dependent. For comparison, we have observed ratios as large as 10:1 for iBGP to eBGP traffic in the AT&T network.

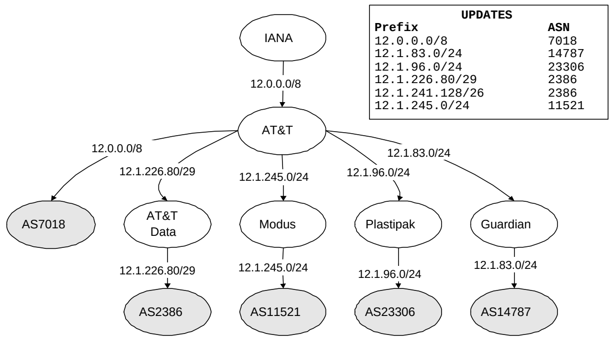
In recognition of the problems inherent of determining a perfect representation, we approximate the delegation graph using available interdomain routing information. The following lists several of the relevant sources and considers the quality of delegation information that they represent.

a) **IANA** - IANA is the origin of all delegation of IP address space. IANA directly delegates address space to 46 unique organizations [12]. 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 every AS announces every address space they are delegated. Any advertisement *encapsulated* (e.g., 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 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.



UPDATES

Prefix	ASN
12.0.0.0/8	7018
12.1.83.0/24	14787
12.1.96.0/24	23306
12.1.226.80/29	2386
12.1.241.128/26	2386
12.1.245.0/24	11521

Figure 2: Address Delegation graph for prefix `12.1.0.0/16`.

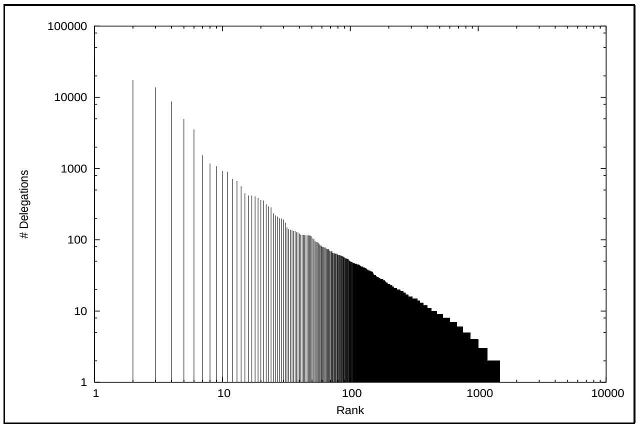


Figure 3: Delegation - cumulative distribution function for the delegation in the approximate delegation graph.

However, we do consider the possible effect of underestimation on our results in section 6.3.

5.2 An Approximate Graph

We have selected (a) IANA and (b) 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, much if not all of the relations between organizations that would be used to inform delegation are reflected in the BGP announcements. The RouteViews project [19] repository is our source of BGP announcement data. The delegation graph integrates public information published by IANA and obtained a single table update from April 1st, 2003. The BGP table contained 129,731 distinct prefixes advertised by 14,912 ASes. Such numbers are consistent with Huston's detailed ongoing evaluations of BGP advertisements [11].

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 `12.0.0.8/8`. 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 a assignment edge to the graph for each such announcement. All other non-self delegations were handled in a similar manner; a delegation edge was added from the parent organization when no encompassing AS advertisement exists. 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 Figure 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&T's AS announced both `12.0.0.0/8` and `12.1.0.0/16`). These longer prefixes were ignored.

The complete graph resulting from the graph approximation process on the data cited above can be viewed at:

`http://www.pdmcdan.com/bgp/delhier.html`

The approximated graph shows that 2,112 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.[7]

In Figure 3 we rank each node according to the number of delegations from that node in the delegation graph and 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 [34]. (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 vs d is given by $n(d) \sim 1/d^{\beta}$ for some constant β [1, 4, 7]. The power law delegation distribution implies the Zipf distribution for number of delegations and we omit the graph of it for lack of space.) The most striking fact shown by this data is that the overwhelming number of delegations are being performed by a relatively few ASes/organizations. In this case, 16 AS/organizations are responsible for 80% of the delegation on the Internet. Furthermore 122 ASes/organizations are responsible for 90% of the delegations and 1,220 perform %99 of the delegations. The top ten delegators are: 1-ARIN (30%), 2-*various registries*[8] (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 improve performance where few authorities provide proofs to arbitrary collections of constituents. We revisit and confirm this via simulation in section 6.2.

5.3 Delegation Stability

The stability of the delegation hierarchy contributes greatly to the performance of origin authentication. If delegation is highly fluid,

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

[8] 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.

Class	**Jan-Feb**	**Feb-Mar**	**Mar-Apr**	**Apr-May**	**Jan-May**	**Jan-May** (*filtered*)
Stable	117117 (90.0%)	116741 (90.1%)	116340 (87.5%)	119701 (89.0%)	103397 (72%)	128350 (89.6%)
Added	5774 (4.4%)	4925 (3.8%)	9667 (7.2%)	5800 (4.3%)	19001 (13.2%)	6977 (4.8%)
Removed	5465 (4.2%)	6207 (4.7%)	4246 (3.1%)	7017 (5.2%)	15770 (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	129988 (100%)	129448 (100%)	132908 (100%)	134462 (100%)	143215 (100%)	143215 (100%)

Table 1: Delegation Stability - worst case stability of the IP address delegation graph from January to May 2003. The filtered data approximates best-case stability of the delegation graph (*see below*).

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 [26]. This section considers if the same is true of the delegation of the IPv4 address space. Notes that this preliminary 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 5 months of 2003. We obtained a single BGP table from the first day of the 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 to February) and over the entire period (e.g., compared January to 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.

This first 5 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. A 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 oscillation 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 can not differentiate a legitimate move from a multi-homed prefix that only oscillates between ASes only 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 effect 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 indicates that the delegation hierarchy evolves slowly, where only about 10% of the delegations (representing 10 to 15 thousand delegations in the worst case) change on any given month. Moreover, as shown by the Jan.-May measurements, the rate of change is relatively constant. The best case analysis exhibits similar properties, albeit at about half the rate of change.

Construction	Sig.	Hash	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	$n \log \frac{m}{n}$	$\phi + n\mu \log \frac{m}{n} + n\alpha$

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

6. EVALUATION

The approaches defined in the preceding section have 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 trade-offs on the consumption of resources (e.g., storage vs. 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 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 ASe/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 is illustrated through the following fictional example. Assume that a signature size is 110 bytes (from [16], $\phi = 110$), four-byte AS/organization identifiers ($\alpha = 4$), and the output of the hash function is 16 bytes (e.g., as per MD5 [27],$\mu = 16$)), 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 by simple attestations is 11400 bytes, 4110 for authenticated delegation lists, 6200 for AS authenticated delegation lists, and 2110 to 8510 bytes for an authentication delegation tree.

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. 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 hour period (midnight to midnight).

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 locally cached (e.g., not considered when calculating cache sizes). Because no AS or non-delegating organization need be issued a certificate, the number of certificates is roughly equal to the number of ownership sources (e.g., 2,112 in this test).

In all tests, we model online operation as transmitting delegation and assignment proofs through the BGP optional transitive attributes [30] . The bandwidth experiments ignore the current BGP MTU (4096 bytes). We seek to understand the efficacy of optimal solutions, and as such relax 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). Because they are uniform, we do not measure costs associated with the acquisition and management of organization certificates.

A first battery of tests makes a broad comparison of the origin authentication methods. Figure 4 shows the computational costs as measured by signatures in 5 minute increments of the 24 hour trace period (for legibility, the figures only show a representative 4 hour 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 significantly less costly – both require an order of magnitude less than simple attestations. Ownership Sources 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 Figure 5, bandwidth costs in online OA are discrete. Authenticated delegation lists are significantly more expensive that 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. No period consumed more than 100k of bandwidth for any construction, and most less than 10k. This stands to reason: very few proofs (10s) are validated in any period. The only exception 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 contains 114,183 delegations. Caching all proofs for these delegations requires 13.4M cache for simple attestations, 1.2M for authenticated delegation lists, 4.0M for AS authenticated delegation lists, and 4.7M for authentication delegation trees.

Figures 6 and 7 show the computational costs associated with each scheme under varying cache sizes over a two hour period (4:40pm-9pm). 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 caches sizes (1M and 100k) are effected by reference locality. The most notable aspect of these graphs is the degree to which aggregate proof schemes (i.e., lists/trees) outperform others (note the vastly different y-axis scale). This is a direct result of the structure of the delegation graph: the 16 proofs encompassing 80% of the delegations are likely to remain in the cache. Non AS-centric approaches make have no such association, and therefore do not take advantage of the delegation density.

The results lead to a new cache strategy for aggregate proof schemes: *caching signatures only*. The cache would maintain a complete cache of signatures for just over 200 kilobytes. This would perform as if all values were previously cached. We will consider these and other strategies (e.g., LFU caching disciplines) in future work.

6.3 What if we are wrong?

Assume that our approximation of the delegation graph is completely wrong: the IP address delegation graph changes frequently and has many nodes of high degree. 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 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

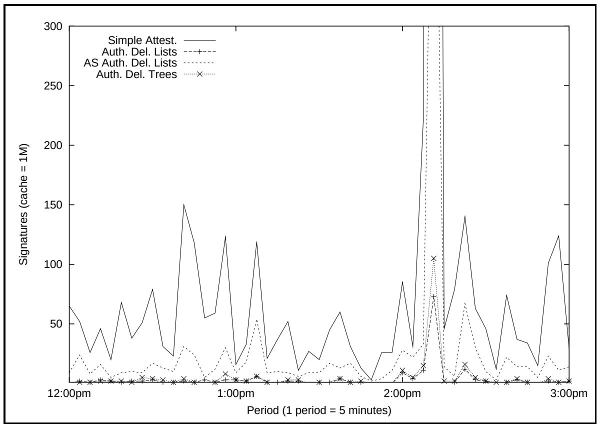


Figure 4: computational cost - signature validations of each origin authentication scheme.

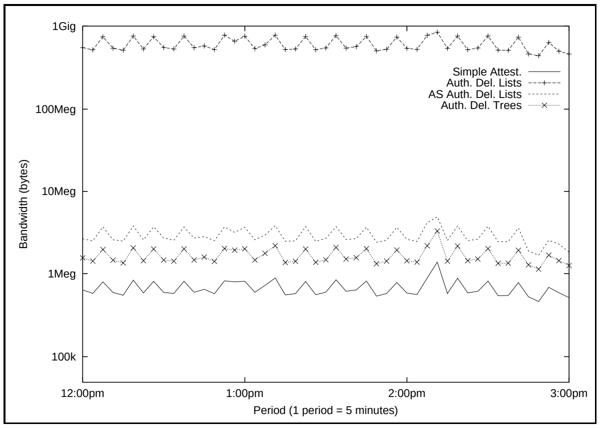


Figure 5: bandwidth cost - bandwidth costs of origin authentication schemes.

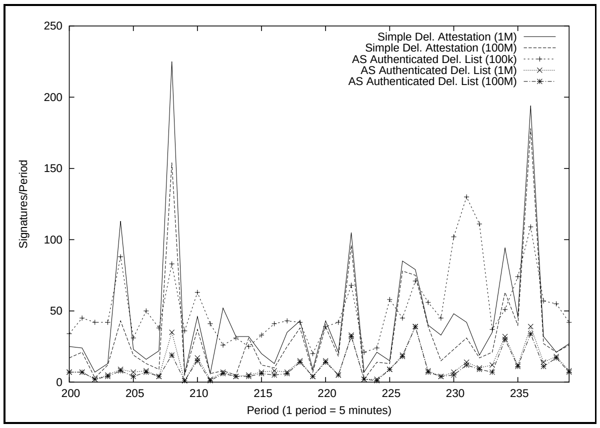


Figure 6: Cache evaluation - signature validations for attestations and AS authenticated delegation lists.

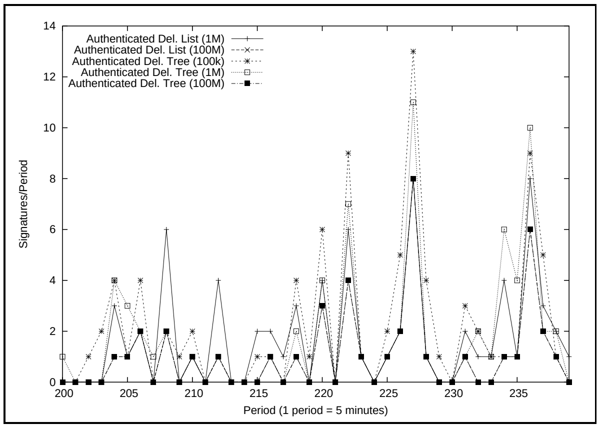


Figure 7: Cache evaluation - signature validations under authenticated delegation lists and trees.

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 effect 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 important 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 which 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 extending existing 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 of these systems show that resource costs can be reduced by up to an order of magnitude over proposed solutions. Such results indicate that on-line origin authentication may now be in 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 was 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 the energy barrier of interdomain routing security can be breached.

8. REFERENCES

[1] W. Aiello, F. Chung, and L. Lu. Random Evolution of Massive Graphs. In *Proceedings of IEEE Symposium on Foundations in Computer Science*, pages 510–519. IEEE, 2001. Las Vegas, Nevada.

[2] W. Aiello, S. Lodha, and R. Ostrovsky. Fast Digital Identity Revocation. In *Proceedings of CRYPTO 98*, pages 137–152, August 1998. Santa Barbara, CA.

[3] ARIN. American Registry for Internet Numbers, May 2003. `http://www.arin.net/`.

[4] A. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286:509–512, 1999.

[5] S. Cheung. An Efficient Message Authentication Scheme for Link State Routing. In *13th Annual Computer Security Applications Conference*, pages 90–98, December 1997. San Diego, California.

[6] R. W. (editor). Deployment Considerations for Secure Origin BGP (soBGP). Internet Research Task Force, October 2002. (`draft-white-sobgp-bgp-extensions-00.txt`).

[7] M. Faloutsos, P. Faloutsos, and C.Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proceedings of ACM SIGCOM Conference*. ACM, 1999. Cambridge, MA.

[8] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and 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, February 2003. San Diego, California. (*Draft*).

[9] B. Green. *BGP Security Update: Is the Sky Falling? NANOG 25*, June 2002.

[10] Y. Hu, A. Perrig, and D. Johnson. Efficient Security Mechanisms for Routing Protocols. In *Proceedings of Network and Distributed Systems Security 2003*. Internet Society, February 2003. San Diego, California.

[11] G. Huston. Bgp table data, February 2003. `http://bgp.potaroo.net/`.

[12] IANA. Internet Protocol V4 Address Space, February 2003. `http://www.iana.org/assignments/ipv4-address-space`.

[13] IANA. The Internet Assigned Numbers Authority, May 2003. `http://www.iana.org/`.

[14] ICANN. The Internet Corporation for Assigned Names and Numbers, May 2003. `http://www.icann.org/`.

[15] S. Kent, C. Lynn, J. Mikkelson, and 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.

[16] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, April 2000.

[17] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP Misconfiguration. In *Proceedings of ACM SIGCOMM '02*. ACM, September 2002.

[18] R. Merkle. Protocols for Public key Cryptosystems. In *Proceedings of the 1980 Symposium on Security and Privacy*, pages 122–133. IEEE, April 1980. Oakland, CA.

[19] D. Meyer. The RouteViews Project, May 2003. http://www.routeviews.org/.

[20] S. Micali. Efficient Certificate Revocation. Technical Report Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, 1996.

[21] S. Misel. Wow, as7007! `http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html`.

[22] S. Murphy. BGP Security Vulnerabilities Analysis (*Draft*). Internet Research Task Force, February 2002. (`draft-murphy-bgp-vuln-00.txt`).

[23] M. Naor and K. Nassim. Certificate Revocation and Certificate Update. In *Proceedings of the 7th USENIX Security Symposium*, pages 217–228, January 1998.

[24] R. Perlman. Network layer Protocols with Byzantine Robustness. Technical Report MIT/LCS/TR-429, October 1988.

[25] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP 4). *Internet Engineering Task Force*, March 1995. RFC 1771.

[26] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. BGP Routing Stability of Popular Destinations. In *ACM SIGCOMM IMW (Internet Measurement Workshop) 2002*, 2002.

[27] R. Rivest. The MD5 Message Digest Algorithm. *Internet Engineering Task Force*, April 1992. RFC 1321.

[28] K. Seo, C. Lynn, and 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.

[29] B. Smith and J. Garcia-Luna-Aceves. Securing the border gateway routing protocol. In *Proceedings of Global Internet '96*, pages 103–116, November 1996.

[30] J. Stewart. *BGP4: Interdomain Routing in the Internet*. Addison-Wesley, 1998.

[31] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet Hierarchy from Multiple Vantage Points. In *Proceedings of IEEE INFOCOM 2002*. IEEE, June 2002.

[32] Z. Wenzel, J. Klensin, R. Bush, and S. Huter. Guide to Administrative Procedures for the Internet Infrastructure. *Internet Engineering Task Force*, August 2000. RFC 2901.

[33] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. An Analysis of BGP Multiple Origin AS (MOAS) Conflicts. In *ACM SIGCOMM Internet Measurement Workshop 2001*. ACM, November 2001.

[34] G. K. Zipf. *Human Behaviour and the Principle of Least Effort*. Hafner, 1949.

APPENDIX

A. APPENDIX A - 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'. $(y/k,\ \mathcal{C},\ \mathcal{N})$ where $\mathcal{C} \subset \mathcal{O}$ and $\mathcal{N} \subset \mathcal{ASN}$.

All the previous options can be captured with this as follows. Option 1., the ASN assignment option, is given by $|\mathcal{N}| = 1$ and $\mathcal{C} = \emptyset$. Option 2., the delegation option, is given by $|\mathcal{C}| = 1$ and $\mathcal{N} = \emptyset$. Option 3., the RESERVED option, is given by $|\mathcal{C}| = \mathcal{N} = \emptyset$. Option 4., the UNAUTHENTICATED option, is given by $|\mathcal{C}| = \mathcal{ASN}$ and $\mathcal{N} = \mathcal{O}$. 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 $\mathcal{C}$ and $\mathcal{N}$. 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 a prefix delegation path.

A more general delegation option still for C is

1". $(y/k,\ \mathcal{Q})$ where $\mathcal{Q}$ 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 will be discussed further in the full paper.