Bootstrapping Accountability in the Internet We Have

Ang Li Xin Liu Xiaowei Yang
Dept. of Computer Science
Duke University
{angl,xinl,xwy}@cs.duke.edu

Abstract

Lack of accountability makes the Internet vulnerable to numerous attacks, including prefix hijacking, route forgery, source address spoofing, and DoS flooding attacks. This paper aims to bring accountability to the Internet with low-cost and deployable enhancements. We present IPA, a design that uses the readily available top-level DNSSEC infrastructure and BGP to bootstrap accountability. We show how IPA enables a suite of security modules that can combat various network-layer attacks. Our evaluation shows that IPA introduces modest overhead and is gradually deployable. We also discuss how the design incentivizes early adoption.

1 Introduction

Accountability, the ability to identify misbehaving entities and deter them from misbehaving further, plays a critical role in achieving real-world security [41]. However, the Internet design has little built-in accountability: malicious hosts can send denial of service (DoS) flooding packets with spoofed source addresses to evade punishment; and malicious Autonomous Systems (ASes) can announce other ASes' IP prefixes or assume their identities in the inter-domain routing system BGP.

Lack of accountability has led to many of the Internet's security vulnerabilities [20, 58], including distributed DoS attacks that may disable a country's Internet access [48, 49, 52], and prefix hijacking attacks that once made YouTube worldwide unreachable [25]. In this work, we ask the question: *can we overcome the Internet's main security weaknesses with a minimal set of gradually deployable changes?* That is, we aim to explore an approach that can fix the Internet's security problems without replacing or breaking the deployed Internet base. We are attracted to this approach because of its practical value, as it can deliver benefits without building everything from scratch.

In this paper, we present a design called IPA (IP made Accountable) that bootstraps accountability in the Internet with only low-cost and gradually deployable enhancements. We show how the IPA design enables other security modules that together fix many of the Internet's security problems, including preventing prefix hijacking, route forgery, and source address spoofing attacks, and limiting large-scale DoS attacks. We note that this work does not aim to provide all forms of accountability. For instance, IPA does not provide the type of strong accountability that offers evidence of correct execution, or audit and challenge interfaces [32, 60]. Rather, it aims to bring a similar form of network-layer accountability as defined in [20, 54] to the Internet, *i.e.*, the ability to accurately identify the sources of all traffic and defend against malicious sources.

We identify two key challenges in bootstrapping accountability in the existing Internet. The first one is how to securely bind an entity's identity to its cryptographic keys in a lightweight manner, and the second one is how to do so in an adoptable manner, including being gradually deployable and incentivizing early adoption. Network-layer accountability requires a secure binding between an entity's identity and its cryptographic keys to prevent impersonation and identity white-washing attacks [31]. The Internet uses two types of identifiers, IP addresses and AS numbers (ASNs), to identify network attachment points and ASes, but it lacks a lightweight and adoptable mechanism to create the secure bindings between an IP address (or an ASN) and a network entity. Previous work [38, 46, 56, 57] proposes to use a centralized global public key infrastructure (PKI) or web-of-trust to bind an IP prefix or an ASN to its owner's public key. However, a dedicated PKI is too heavyweight [35], and web-of-trust lacks an authoritative trust chain to resolve conflicting IP prefix or ASN claims.

IPA uses three mechanisms to address these challenges. First, it uses the top-level reverse DNSSEC hierarchy as a lightweight PKI to bind an IP prefix to its owner's public key (§ 3.2), and the hash of an AS's public key as its self-certifying ASN (§ 3.1). This design securely certifies an IP prefix's ownership without a separate PKI, and obviates another PKI to certify an ASN's ownership. We use DNSSEC [21, 22, 23, 50] because one can create a one-to-one mapping between an IP prefix delegation and a reverse DNS zone delegation, as the chains of trust in both delegation processes share the same root: the Internet Assigned Number Authority (IANA). Thus, we can use an IP prefix's corresponding reverse DNSSEC record as its owner's IP prefix delegation certificate. Moreover, Internet registries are rapidly deploying the top-level reverse DNSSEC infrastructure [4, 6, 18, 19]. The root, the `arpa`, and the `in-addr.arpa` zones are already signed. Deployment documents from key Regional Internet Registries (RIRs) [1, 2, 5] all suggest that the top-level reverse DNSSEC infrastructure would soon be fully deployed.

Second, IPA uses an efficient in-band protocol piggybacked in BGP messages to "push" the IP prefix certificates to all ASes to secure routing (§ 3.3). This design avoids the dependency loop between secure routing and online certificate distribution, and eliminates the need for a separate out-of-band certificate distribution mechanism. We strive to make the in-band distribution protocol efficient and capable of supporting complex operations such as certificate revocations and key rollovers (§ 4).

Third, we design IPA to be compliant with the existing protocols to be gradually adoptable. It uses the BGP optional and transitive attributes to carry IPA-specific information so that legacy ASes can pass this information to deployed ASes without interpreting them (§ 7.3.1). Different ASes can deploy IPA at different times without a "flag day." Furthermore, because we use the top-level reverse DNSSEC hierarchy to bind IP prefixes to their owners' public keys, the ASes who obtain their IP prefixes from the Internet registries can obtain their prefix ownership certificates from the registries without depending on other infrastructures. This feature enables those ASes, which amount to 78% of all ASes on today's Internet (§ 7.3.2), to form a deployed "club" to prevent various network-layer attacks within the club (§ 5).

We further show how IPA enables several security building blocks, including a secure routing protocol such as S-BGP [38], a source authentication system [43], and a DoS defense system [45] (§ 5). These security building blocks are also gradually adoptable [26, 43, 45], and together can prevent prefix hijacking, route forgery, and source address spoofing attacks, and suppress DoS flooding traffic near its sources.

We have implemented IPA using XORP [33] and integrated other security modules with it (§ 6). We evaluate IPA's performance and adoptability using trace-driven experiments (§ 7.2), live Internet experiments (§ 7.3.1), and analysis (§ 7.3.2). The results suggest that IPA is lightweight and gradually deployable in the current Internet. Our trace-driven experiments show that IPA's query overhead on an Internet registry's DNS servers is less than 0.1% of a single root DNS server's regular workload. Its in-band certificate distribution protocol introduces modest overhead to a router. A single-threaded IPA implementation running on a commodity PC can process all messages a RouteViews server [53] receives at their arrival rate. We expect that the server's workload is representative of a large ISP's BGP router's workload, because the number of peers it has (37) is the top 6% largest among all ASes [8].

Our live Internet experiments show that IPA's protocol messages piggybacked in BGP can pass standard-compliant legacy routers. Our analysis suggests that IPA lowers the deployment cost for early adopters compared to previous work that requires dedicated PKIs [38, 46, 56, 57], but offers equivalent or stronger security strength. Thus, it is more likely to be adopted.

To the best of our knowledge, IPA is the first design that brings accountability to the Internet in a secure, lightweight, and gradually adoptable manner.

2 System Models and Goals

Before we present the IPA design, we first describe its system models and design goals.

2.1 System Models

Network Model: IPA adopts the same two-level hierarchical network model (nodes and ASes) as the present Internet. For inter-AS routing and forwarding, we treat an AS as one trust and fate-sharing unit. AS boundaries are also trust boundaries. For clarity, we abstract each AS as a node when describing AS-level operations.

Trust Model: IPA assumes the same external trust entities as the present Internet. The global root of trust is the Internet Assigned Numbers Authority (IANA).

Threat Model: We assume that both hosts and routers can be compromised. Compromised nodes (hosts or routers) can collude into groups and launch arbitrary attacks. We also assume that an AS may be malicious, and malicious ASes can also collude.

2.2 Design Goals

IPA's central design goal is to securely bootstrap accountability in the Internet with lightweight and adoptable enhancements. We elaborate it in more detail.

Secure: IPA aims to enable cryptographically provable network-layer identities. As we show in § 5, this ability further enables various security modules that can prevent prefix hijacking [34, 38], route forgery [34, 38], source address spoofing [43], and DoS flooding attacks [45].

Lightweight: We aim to introduce only lightweight enhancements to the Internet. We believe that enhancing the existing infrastructures with new functions has lower deployment costs than rolling out new global infrastructures. For this reason, IPA does not require new global infrastructures, unlike [12, 38, 57]; nor does it require trusted hardware at end systems (although it can help), unlike [20]. Moreover, we aim to add little performance overhead to the deployed Internet base.

Adoptable: We aim to make IPA adoptable, which implies two sub-goals:

- **Gradually Deployable:** We aim to make IPA compatible with the legacy Internet and ready to be deployed on the Internet. IPA-enabled ASes (or hosts) should be able to run IPA-related protocols even if they are connected by legacy ASes.

- **Incentivizing Early Adoption:** IPA should require low deployment costs and provide immediate security benefits to early adopters to incentivize deployment. That is, the group of early adopting ASes should gain security benefits within the deployed region without requiring other entities outside the group to deploy IPA.

3 Overview

This section presents a high-level overview of IPA. We present more design details in the following section. IPA uses two key mechanisms to be lightweight and gradually deployable: 1) it uses the top-level reverse DNSSEC infrastructure as a lightweight PKI to bind an IP prefix to its owner's public key; and 2) it uses the BGP routing system to distribute IP prefix certificates in-band.

3.1 A Hybrid Approach to Secure Identifiers

The present Internet uses two types of identifiers: 1) a hierarchically allocated IP address (or prefix) to loosely identify a network attachment point (or a group of them in the same network), and 2) a flat AS number to identify an autonomous system. IANA is the root of trust and the owner of all IP addresses, *i.e.*, the owner of `0/0`. It delegates sub-prefixes to RIRs, which in turn delegate even smaller sub-prefixes to ASes. ASes may further sub-delegate IP prefixes to their customers. Figure 1 shows an example of the address delegation hierarchy.

To be gradually deployable, IPA retains the hierarchical structure of IP addresses, and uses the existing chain of trust in the IP address allocation process to bind an IP prefix to its owner's public key. Since ASNs do not have a hierarchical structure, IPA replaces them with ASes' self-certifying identifiers, *i.e.*, the hash of their public keys. This design reduces the deployment overhead at an Internet registry, as a registry need not bind an AS's identifier to its public key. This new ASN format can be gradually deployed in a manner similar to how the 32-bit ASN was recently deployed [55].

3.2 DNSSEC as a Lightweight PKI

The IPA design uses the top-level DNSSEC infrastructure as a lightweight PKI for Internet registries to issue IP prefix delegation certificates. DNSSEC is originally designed to protect the integrity of DNS replies. Similar to a PKI, it allows a parent entity to use its key to certify a DNS zone delegation to a child entity. Each zone owner signs the DNS records in its zone, and publishes their signatures in DNS for verification. When a client performs a DNSSEC query for a domain name, it can verify the authenticity of the answer by following the DNS hierarchy to obtain the relevant DNSSEC records.

Using DNSSEC to certify IP prefix delegation has several advantages. First, we can create a one-to-one mapping between a reverse DNS zone delegation and an IP prefix delegation, as the reverse DNS hierarchy and the IP address hierarchy share the same root (IANA). For example, when IANA delegates an IP prefix `165/8` to an RIR (ARIN), it can also delegate the corresponding reverse DNS zone, `165.in-addr.arpa`, to ARIN (Figure 1). This delegation further enables ARIN to create a one-to-one mapping between the IP sub-prefixes and the reverse DNS zone's sub-delegations, *e.g.*, delegating `165.72/16` and `72.165.in-addr.arpa` to an AS (AT&T). A prefix owner can use the DNSSEC records that certify its reverse DNS zone delegation as a certificate authorizing its prefix ownership (§ 4.1). We refer to this type of certificate as an *IP prefix delegation certificate* or a *prefix certificate*. This design reduces IPA's deployment costs at an Internet registry, as it need not maintain a separate PKI to certify IP prefix delegations.

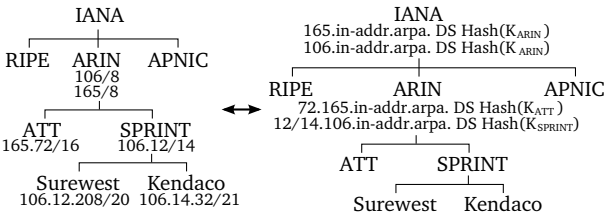

Figure 1: ***Left*: the IP prefix allocation hierarchy; *Right*: the corresponding DNSSEC records that bind the prefixes to their owners' public keys.**

The second advantage is that Internet registries are rapidly deploying DNSSEC [7, 29, 50]. The root zone was signed in July 2010 [19], and later the `arpa` and the `in-addr.arpa` zones. IANA will further sign the sub-zone delegations from `in-addr.arpa` in late March 2011 [7]. Moreover, the three largest RIRs, ARIN, RIPE, and APNIC, have all stated in their websites that they are ready to or will soon be ready to sign reverse zone sub-delegations [1, 2, 5]. Since these RIRs own 142 out of 175 sub-zones of `in-addr.arpa` [14], we expect that the top-level reverse DNSSEC will soon be fully deployed by all Internet registries.

Finally, because DNSSEC supports online queries, an Internet registry can use it to publish new IP prefix certificates to support key rollovers (§ 4.5) or revocations (§ 4.2), in addition to issuing certificates. An AS can query the DNS to download its up-to-date prefix certificates and the Internet registries' revocation lists.

3.2.1 IP Prefix Sub-delegation

After an AS obtains its IP prefixes, it may delegate sub-prefixes to its customers. For instance, Sprint in Figure 1 allocates a sub-prefix `106.12.208/20` to its customer Surewest. The IPA design allows an AS to flexibly choose the infrastructure it uses to manage these sub-delegation certificates. An AS can choose to use DNSSEC, as does an Internet registry. Alternatively, it

may use a certificate authority server to issue the IP prefix certificates. In the latter case, an AS should also support a certificate publishing mechanism (*e.g.*, a secure web server or an FTP server) to enable its customers to download their up-to-date certificates online. This requirement is to support automatic key rollovers (§ 4.5). We believe that an AS has incentives to manage and publish its customers' certificates, because this effort can protect its customers from prefix hijacking attacks.

For clarity, in the IP prefix delegation process, we refer to the delegator as the *parent owner*, and the delegatee as the *child owner*.

3.3 In-band Certificate Distribution

To prevent routing attacks, ASes must use a secure routing protocol (*e.g.*, S-BGP [38], § 5.1) to validate prefix origins and AS paths in BGP messages. This requires ASes to first obtain valid IP prefix certificates.

IPA uses BGP itself to distribute these certificates in-band to ASes that need them. That is, when an AS originates an IP prefix in a BGP message, it piggybacks the chain of certificates that can prove its prefix ownership in the message. We use a BGP feature, the transitive and optional path attribute, to carry the certificates. An AS can first obtain the chain of certificates offline when it obtains the IP prefix from its parent AS or an Internet registry. Later, it can periodically download the full chain of the latest certificates, as we will describe in § 4.5.

This design has several advantages. First, it avoids the dependency loop between secure routing and online certificate distribution. If we use an alternative approach where each AS downloads the prefix certificates from online distribution servers (*e.g.*, DNSSEC servers), a dependency loop between routing and certificate distribution may occur. This is because to obtain a prefix p's certificate C_p, an AS X must first establish a valid path to an AS Y that hosts C_p's distribution server. Recursively, to establish a valid path to AS Y, X must validate the BGP messages advertising AS Y's prefixes, which requires AS X to have obtained AS Y's prefix certificates. These certificates may be served by a distribution server in yet another AS Z, and to establish a valid path to Z, X needs the certificates for Z's prefixes, and so on. These dependencies may eventually form a loop, preventing AS X from obtaining the certificates needed to validate the prefix p's ownership.

In contrast, in-band distribution does not introduce such dependencies. This is because it does not require an AS to establish an *a priori* valid path to an online distribution server. BGP messages are propagated hop-by-hop (at the AS level). An AS will first obtain valid certificates from its neighbors, and then from its neighbors' neighbors, and so on, until it obtains the valid certificates from all ASes in the routing system.

Second, in-band distribution lowers deployment costs, as it does not need an out-of-band channel to distribute the certificates, unlike [38, 56]. IPA also uses standard BGP features to encode the certificates so that different ASes may gradually adopt the distribution mechanism without breaking BGP.

Finally, including a prefix p's full chain of certificates ensures that any AS that receives a BGP message originating p can immediately validate p's owner's public key. This further ensures that an AS can promptly validate the prefix origin and AS path in the BGP message (§ 5.1) and propagate the message and the chain of certificates further to its neighbors. These neighbors can in turn use the certificates to validate the BGP message and propagate it further, until all ASes have received and validated the BGP message. We refer to this property as *liveness*, and provide a formal proof of it in [42]. We discuss how to validate a certificate in § 4.4.

Attaching a full chain of certificates in a BGP message incurs significant communication overhead. IPA uses a simple but effective technique to reduce this overhead: each AS caches the certificates that it has sent to a neighbor and only sends to the neighbor the certificates that it has not sent yet. We describe it in more detail in § 4.3.

4 Design Details

This section presents more design details of IPA, including how to use DNSSEC records to encode an IP prefix certificate (§ 4.1), certificate revocation (§ 4.2), efficient certificate distribution (§ 4.3), certificate validation (§ 4.4), and key management (§ 4.5).

4.1 DNSSEC Records as IP Prefix Certificates

IPA uses three types of a reverse DNS name's resource records to encode a prefix certificate: the designated signer (DS) record, the public key (DNSKEY) record, and the signature (RRSIG) record of the DS record.

Figure 2 shows the DNSSEC records that form the certificate for the prefix `165/8`, which IANA allocates to ARIN (Figure 1). These records are associated with the DNSSEC entry `165.in-addr.arpa` created by IANA. IANA uses the DS record to store the hash of ARIN's pubic key, and signs the DS record using its private key. It sets the inception and expiration times of the signature record (RRSIG) to the inception and expiration times of the prefix allocation, and publishes the entry `165.in-addr.arpa` on its DNS servers. This process follows the standard DNSSEC practice, and also applies to IPv6 address allocation.

A slight complication arises as not all IP address allocations fall on a reverse DNS domain boundary. For instance, as shown in Figure 1, ARIN may allocate an IP prefix `106.12/14` to Sprint. We address this issue by extending the encoding format of a re-

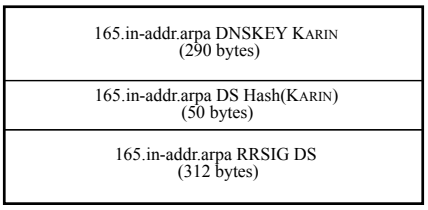

Figure 2: **This figure shows the DNSSEC records that encode the prefix `165/8`'s certificate. The size of each record is estimated assuming that the signatures are generated using 2048bit RSA/SHA-1.**

verse DNS name. For instance, we use the reverse DNS name `12/14.106.in-addr.arpa` to encode the IP prefix `106.12/14`. The encoding/decoding rules are straightforward and compatible with the DNS standard [47]. We omit them due to the lack of space, but describe them in [42]. We choose not to use the existing techniques that support classless reverse zone delegations [27, 30], because they either only support allocations in chunks smaller than a `/24` prefix [30], or are no longer supported by popular DNS servers [9, 27].

4.2 Revoking an IP Prefix Certificate

An Internet registry or an AS may revoke a certificate allocated to a child before it expires. This may occur if the prefix is re-assigned to a new child owner, or the child owner's key is compromised, or the child owner violates the terms of use or switches to a different ISP.

In the IPA design, a parent owner issues a new prefix certificate to explicitly revoke the old one. The new certificate binds the IP prefix to a new public key with a newer inception time. The new key could be a new child owner's key, or the present child owner's new key, or the parent's own key if it reclaims the IP prefix from a child.

As we discuss in § 3.3, IPA distributes IP prefix certificates in the routing system for ASes to validate routing messages. To use a certificate to validate a routing message, an AS must know whether the certificate has been revoked or not. IPA uses both *push* and *pull* mechanisms to notify an AS of a certificate's revocation status.

Pushing New Certificates via Routing: Because a new certificate explicitly revokes an old one, a new certificate's owner can immediately announce the new certificate in BGP using the in-band distribution mechanism to notify other ASes of the old certificate's revocation.

Periodic Pulling From Internet Registries: When an Internet registry revokes a prefix certificate, the registry may be unable to notify other ASes using the push-based mechanism, because it does not participate in routing. We use a DNSSEC-based revocation list to address this problem. A revocation list includes the set of IP prefixes an Internet registry reclaims from its children, or re-assigns to its children that are also Internet registries. The registry can publish the list using a TXT record with a special DNS name, *e.g.*, `revoked.arin.in-addr.arpa`, and sign the list using DNSSEC. An entry in a revocation list includes the revoked IP prefix and the revocation time. It revokes any older prefix certificate signed by the same registry and whose address range overlaps with the revoked prefix.

Each AS periodically (*e.g.*, daily) downloads the revocation lists from all Internet registries to invalidate revoked certificates (§ 4.4). An AS does not query DNS at the certificate validation time to reduce DNS load. Periodic downloads may delay a certificate's revocation, but we consider this delay acceptable, as it will not lead to prefix hijacking attacks. Only the IP prefixes not allocated to any AS will suffer this delay, as an AS that owns an IP prefix can immediately announce its new certificate in BGP to revoke the old one.

4.3 Efficient Certificate Distribution

As we describe in § 3.3, IPA uses a BGP message itself to distribute the full chain of certificates of the IP prefix that the message advertises. We now describe how to make this in-band distribution protocol efficient.

Each AS maintains several certificate caches to record what it has sent to a neighbor and to maintain certificate validation state, as shown in Figure 3. The caches include: 1) an incoming certificate cache that stores all certificates received from its neighbors; 2) a trusted certificate cache that stores the certificates it has validated; and 3) a per-neighbor outgoing certificate cache that records the hash of each certificate it has sent to the neighbor. An AS organizes the certificates in its trusted cache in a tree-like structure following the IP allocation hierarchy to assist certificate validation (§ 4.4).

When an AS receives a prefix certificate from a neighbor, it first stores the certificate in its incoming cache, and then validates the certificate as we describe next. When the AS sends a BGP message to a neighbor announcing the IP prefix, it will retrieve the full chain of certificates from its trusted certificate cache, and compare them with those in the neighbor's outgoing certificate cache. It will only send the certificates that are not in the neighbor's outgoing cache, and then insert them in the outgoing cache to avoid sending them to the neighbor again.

When an AS loses the peering connection to a neighbor, *e.g.*, due to a router reboot or link failure, it will remove all entries in the neighbor's outgoing cache. When the AS resumes its connection with the neighbor, it will re-send the full chain of certificates for each prefix it announces to the neighbor.

4.4 Validating IP Prefix Certificates

When an AS receives a BGP message that advertises a prefix p_n and includes a list of certificates from a neighbor, it must validate these certificates to verify p_n's

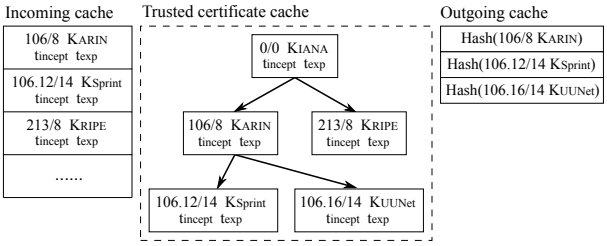

Figure 3: **An example of the certificate caches an AS maintains. It shows only one outgoing cache of the AS.**

owner's public key. It considers a prefix p_n's certificate C_{p_n} valid if C_{p_n} meets the following conditions:

1. C_{p_n} is not on any Internet registry's revocation list or revoked by a newer certificate (§ 4.2).
2. C_{p_n} has a valid parent certificate $C_{p_{n-1}}$ such that 1) C_{p_n} is signed by its parent certificate $C_{p_{n-1}}$'s private key; 2) p_n is a subset of its parent certificate's prefix p_{n-1}. If p_n is the prefix `0/0`, C_{p_n} need not have a parent but must be self-signed by IANA.

Algorithm 1 shows the pseudo-code for the validation algorithm. Most steps of the algorithm check whether C_{p_n} satisfies the above conditions. We note two things. First, if C_{p_n} does not have a valid parent certificate, C_{p_n} becomes unverifiable. Unverifiable certificates may exist temporarily during a key rollover event (§ 4.5). The algorithm returns failure but leaves C_{p_n} in the incoming cache, as it may become valid later after its parent certificate has arrived. Second, the last section of the code (line 23–26) adds the newly validated C_{p_n} to the AS's trusted cache and checks whether any previously unverified certificate C_i is now verifiable, which may happen if C_{p_n} is its parent. If such a certificate C_i exists, the algorithm recursively validates it and its child certificates.

Algorithm 1 **validate**(C_{p_n}): pseudo-code to validate the certificate C_{p_n} in an incoming BGP update message msg.

Input: C_{p_n}, the incoming certificate to be validated; p_n, the prefix of C_{p_n}; msg, the incoming BGP message; $cache_{tr}/cache_{in}$, the current trusted/incoming certificate cache; $rlist[r]$, the most recent revocation list of registry r

```
1:  if is_registry(C_{p_n}.signer)
    and p_n ∈ rlist[C_{p_n}.signer] then
2:      cache_in.remove(C_{p_n})
3:      return false
4:  end if
5:  C_{p_{n-1}} ⇐ cache_tr.lookup_parent(C_{p_n})
6:  if C_{p_{n-1}} == NULL then
7:      C_{p_{n-1}} ⇐ msg.lookup_parent(C_{p_n})
8:      if C_{p_{n-1}} == NULL or not validate(C_{p_{n-1}}) then
9:          return false
10:     end if
11: end if
12: for C_s ∈ cache_tr.get_children_certs(C_{p_{n-1}}) do
13:     if overlap(p_n, p_s) then
14:         if C_{p_n}.inception > C_s.inception then
15:             cache_tr.recursive_remove(C_s)
                // remove all certificates in C_s's subtree
16:             cache_in.remove(C_s)
17:         else
18:             cache_in.remove(C_{p_n})
19:             return false
20:         end if
21:     end if
22: end for
23: cache_tr.insert(C_{p_n})
24: for C_i ∈ cache_in and C_i ∉ cache_tr do
25:     validate(C_i)
26: end for
27: return true
```

4.5 Key Management

Like any cryptography-based system, IPA's accountability builds on the secrecy of private keys. In addition to the standard practice to protect secret keys, IPA takes two additional measures: 1) separating an AS's identity keys from the keys the AS uses to sign routing messages, and 2) periodic key rollovers.

4.5.1 Separating Identity Keys from Routing Keys

To secure routing, an AS must store its private key online to sign routing messages (§ 5.1). Yet it is desirable to keep a private key offline to reduce the risk of key compromise. To balance security and functionality, IPA separates an AS's identity keys from the keys it uses to sign routing messages. We refer to the pair of keys associated with an AS's self-certifying identifier as its identity keys, or its identity key when we refer to either the AS's private or public key.

An AS generates a separate pair of public/private keys to sign routing messages. We refer to this pair of keys as an AS's routing keys. For each IP prefix it owns, an AS will use its identity key to sign a routing certificate that binds the IP prefix to its routing key. The AS keeps its identity private key offline, and uses its routing private key to sign routing messages. An AS will include a prefix's routing certificate in its BGP messages. Other ASes can validate it using the algorithm described in § 4.4.

4.5.2 Routing Key Rollover

By separating identity keys from routing keys, an AS can periodically expire its routing keys, issue new ones, and sign its new routing certificates with its identity key, all without changing its identifier, or re-signing its prefix sub-delegation certificates.

4.5.3 Identity Key Rollover

An entity should also change its identity keys periodically to improve security. To change its identity keys, an entity must 1) request new certificates from its parents, 2) revoke its old certificates, and 3) re-sign each child certificate with its new private key. As can be seen, this process is more complicated than routing key rollover. Thus, an entity should change its identity keys at a lower frequency than its routing keys.

A key challenge we face is how to make a child certificate remain valid throughout a parent key rollover event so that other ASes can verify the child's routing messages. We address this challenge by "pre-releasing" a child's new prefix certificate, a technique similar to how DNSSEC manages key rollovers [39]. With this mechanism, both a child's old and new certificates remain valid during a key rollover event.

For clarity, we first describe the identity key rollover process for an AS, and then for an Internet registry. Figure 4 shows this process. Let D be an AS that wishes to rollover to a new identity key K_{new}. D will first use its old key K_{old} to generate a *transient* certificate certifying K_{new} for each prefix it owns. The transient certificates are only available during key rollovers, and will expire afterwards. Meanwhile, D generates a new certificate for each sub-prefix it delegates to a child using its new key K_{new}. D will also generate new certificates to certify its routing keys using K_{new}. At this point, both K_{old} and K_{new} are valid identity keys of D, because each of them can be certified by a valid chain of certificates, as shown in Figure 4(b). D will then publish the child certificates signed using its new key K_{new} via its certificate publishing system as described in § 3.2.1.

Each AS will periodically (*e.g.*, once a day) query its certificate issuers' publishing systems to download its latest chains of certificates. If the AS obtains IP prefix allocations directly from an Internet registry, it will query the corresponding reverse DNS names of its IP prefixes starting from the root servers. Otherwise, the AS queries its parent ASes' certificate publishing systems. This online certificate downloading step does not have a dependency loop with routing, because each AS's old certificate chain is already in the routing system, and can be used to establish valid paths. If an AS C downloads a new certificate signed by its parent D's new key, it will immediately announce its new certificate in BGP. Other ASes will consider C's new prefix certificate valid, because it is certified by a valid chain of trust, including the link provided by the parent D's self-signed transient certificate, as shown in Figure 4(b).

Finally, the rekeying AS D requests each of its parents P that has delegated an IP prefix to its old key K_{old} to issue a new certificate to its new key K_{new}, after waiting for a long enough period d. The waiting period d should

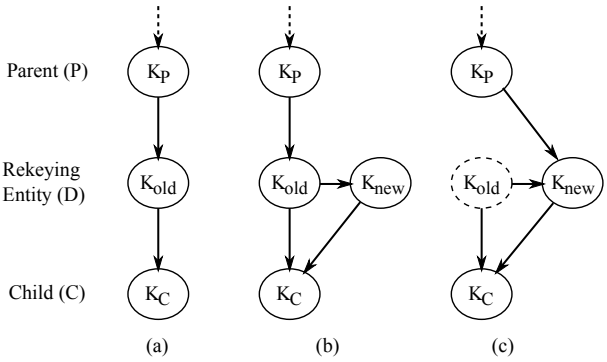

Figure 4: **This figure shows IPA's key rollover process. Each node represents a key; an arrow points from a parent's signing key to a child's signed key. Figure (a) shows the chain of trust before the key rollover; (b) shows the chains of trust during the key rollover, where the rekeying entity D signs a transient certificate to certify its new key K_{new} using its old key K_{old}; (c) shows when the key rollover process finishes, the old key K_{old} becomes invalid.**

be long enough to ensure that each child AS of D has successfully downloaded and announced its new certificates in BGP. D can then announce its new certificate for its new key K_{new} in BGP to revoke its old certificate. The child AS C's certificate will remain valid, as shown in Figure 4(c). An AS D will also re-send its BGP routes to its neighbors using its new identifier.

An Internet registry's key rollover procedure is similar, except that the registry need not announce a new certificate in BGP, as its children will obtain it via DNSSEC.

4.5.4 Recovering From Key Compromise

With the preventive measures we describe above, we expect key compromise to be a rare event in IPA. For completeness, we briefly describe how to recover from it and leave the details to [42].

Recovering from key compromise resembles a key rollover event, except that an entity may resort to contacting its parents and children offline to obtain its new certificates and distribute its children's new certificates. This is because when an attacker compromises an entity's identity keys, it may also hijack the entity's IP prefixes, making it unreachable online.

5 Use of IPA

In this section, we describe how IPA enables various security modules that collectively achieve accountable routing and forwarding, and DoS attack mitigation. Each of the modules we describe here is also gradually adoptable [26, 38, 43, 45].

5.1 Accountable Routing

IPA enables secure routing protocols such as S-BGP [38], because it provides ASes with the necessary

certificates to achieve origin authentication and AS path authentication.

Origin Authentication: An AS O that owns a prefix p can now sign its BGP messages when it announces the prefix, because other ASes can use the chain of certificates piggybacked in the BGP messages to verify the secure binding between the prefix p and O's public key (§ 3.3), preventing other ASes from originating p.

AS Path Authentication: Each transit AS can sign a BGP update using its private key when it prepends its self-certifying AS identifier to the update and propagates the update to a neighbor. A malicious AS cannot forge another AS's identifier, nor can it truncate the AS path, because it cannot generate a valid signature of another AS. A transit AS can piggyback its public key in a BGP message similar to how IPA distributes prefix certificates (§ 3.3). We can also apply the same caching technique described in § 4.3 to reduce the message overhead.

Self-certifying ASNs prevent path forgery, but raise a different security concern: an AS may mint arbitrary identifiers, which complicates BGP policy configurations. The IPA design addresses this concern by binding a self-certifying ASN to an IP prefix. If an AS path contains an ASN that is not a hash of a public key found in a valid IP prefix certificate, other ASes can consider the path not trustworthy, and configure their BGP policies to avoid this path. Moreover, an AS can use IP prefixes to configure its BGP policies, because other ASes cannot arbitrarily change their IP prefixes.

5.2 Accountable Forwarding

The ability to securely sign BGP messages enables Passport [43], a system that can achieve both packet source authentication and forwarding path inconsistency detection. Passport uses a distributed Diffie-Hellman key exchange piggybacked in BGP to establish a shared secret between every pair of ASes. With IPA, an AS O can sign the BGP messages that originate both its prefixes and its Diffie-Hellman public value. Other ASes can securely bind the secrets they share with AS O with O's prefixes to enable AS-level packet source authentication and path inconsistency detection.

Packet Source Authentication: To authenticate a packet's source address, a source AS stamps a sequence of message authentication codes (MACs) into a packet header using the secret keys it shares with each AS en route to the packet's destination. ASes along the path can re-compute the MACs to validate the packet's origin AS, as packets with spoofed source addresses will not have valid MACs.

Forwarding Path Inconsistency Detection: A malicious AS may attempt to advertise one legitimate AS path but forward packets along a different one that conflicts with a source AS's routing policies. The MACs that a source AS stamps into a packet header can help detect this misbehavior. This is because if a packet's forwarding path differs from the AS path its source AS selects to use, an AS on the path will detect an invalid MAC, but the destination AS will detect a valid one. A destination AS can use this discrepancy to notify the source AS of the forwarding path inconsistency.

5.3 DoS Attack Mitigation

Finally, because IPA enables source authentication, it also enables DoS defense systems that use authentic source addresses to suppress attack traffic near its sources, *e.g.*, a filter based system StopIt [44], or NetFence [45], a system based on unspoofable congestion policing feedback.

As an example, we describe briefly how NetFence can use IPA to suppress DoS flooding traffic near its sources. NetFence introduces a secure congestion policing framework in the network. A NetFence packet carries unspoofable congestion policing feedback in a shim layer. An on-path AS updates this feedback to notify an access router of its local congestion conditions, and an access router uses this feedback to regulate a sender's sending rate. The on-path AS and the source AS use the secret they share via Passport to protect this feedback from being tampered by malicious routers or end systems. When malicious sources and receivers collude to flood a link in the network, NetFence provides a legitimate sender its fair share of bandwidth. When a receiver is an innocent DoS victim, NetFence enables the receiver to use the unspoofable congestion feedback as network capabilities [59] to suppress the bulk of unwanted traffic.

We introduce AS-level hierarchical accountability to NetFence to accommodate IPA's self-certifying ASNs. The original NetFence design uses AS-level queues at a router to hold each source AS accountable for its traffic. With IPA, we use hierarchical queuing [24] that follows the IP allocation hierarchy to hold each AS accountable. That is, the traffic from all IP prefixes allocated to an AS's public key will share one queue; a router may subdivide the queue into multiple lower-level queues, if the AS delegates sub-prefixes to its customers, and so on. A router sets a queue's weight according to the size of the IP prefixes associated with the queue, not by the number of ASes sharing the IP prefixes. This mechanism prevents an AS from gaining unfair network resources by dividing its IP prefixes into many smaller ones and delegating them to minted identifiers.

6 Implementation

We have implemented a prototype of IPA's in-band certificate distribution mechanism (§ 3.3) using XORP [33]. The implementation includes a standalone C++ library

`libipa` that other BGP implementations can use. The library `libipa` implements certificate distribution and validation, and supports downloading revocation lists and new certificates from DNSSEC.

Our implementation addresses several practical issues that arise when an IPA router peers with a legacy router. First, we disable the optimization technique (§ 4.3) on an IPA router's interface facing a legacy router, because a legacy router does not cache any certificate or public key. Furthermore, legacy BGP has a 4KB limit on the size of an update message. To bypass this limitation, an IPA router breaks a message longer than 4KB into smaller ones, each of which carries a subset of the certificates and public keys of the original message. The router sends them in sequence to its legacy neighbor. The IPA router waits for a period of time longer than the BGP's MRAI timer (*e.g.*, a few minutes) between sending out two consecutive messages to prevent the first message from being overwritten by the second one.

We have also extended previous implementations of S-BGP, Passport, and NetFence and incorporated them into the IPA prototype. We defer a systematic evaluation on the integrated architecture to future work.

7 Evaluation

In this section, we evaluate IPA along four dimensions. First, we use small-scale testbed experiments to validate the design and implementation. Second, we use trace-driven benchmarks to measure the design's performance and overhead. Third, we use live Internet experiments and analysis to evaluate the design's adoptability. Finally, we analyze IPA's security properties.

7.1 Testbed Experiments

We use DETERlab [28] experiments to validate the design and implementation of IPA. These experiments include 1) bootstrapping experiments, 2) key rollover experiments, and 3) prefix hijacking experiments. We sample a small test topology from the AS-level Internet topology inferred from BGP table dumps. This topology includes six university ASes and all ASes on the shortest AS paths between the six ASes. It contains 17 ASes and 54 uni-directional links. We desire to run larger-scale experiments, but are limited by the number of testbed machines we can obtain. For simplicity, we assume each AS owns one prefix, and choose the prefix to be the largest one the AS owns in reality. Finally, we assume all ASes use DNSSEC to issue and publish their certificates, and use the signing tool included in BIND9 [3] to generate the certificates. The topology includes four levels of IP prefix allocation: IANA, RIRs, top-level ASes, and customer ASes. We randomly pick three ASes to host the root and two RIRs' DNSSEC servers. We assume each AS's DNSSEC server is inside its network. Each node in a testbed experiment corresponds to an AS. Each AS is configured with an initial IP prefix certificate chain.

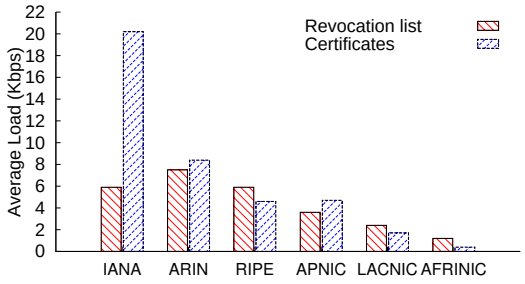

Figure 5: **This figure shows the average DNS traffic load of each Internet registry to serve the revocation list and the IP prefix certificates.**

We summarize the testbed experiment results as follows. In a bootstrapping experiment, each node can validate all certificates and store them in its trusted cache, suggesting that the system can successfully bootstrap, consistent with the liveness property of IPA's in-band certificate distribution protocol (§ 3.3). In a key rollover experiment, the rekeying ASes can successfully propagate their new certificates, and each prefix always has at least one valid chain of certificates during the rollover period. Finally, we run our S-BGP module using the certificates distributed by IPA. We launch a prefix hijacking attack from an AS. All other ASes reject the update message because there does not exist a certificate chain certifying the AS's ownership of the hijacked prefix.

7.2 Performance

IPA adds overhead to both DNS and BGP. We use trace-driven benchmarks to evaluate this overhead. The results show that IPA's overhead on DNS and BGP is acceptable. We use a PC with Xeon 3GHz CPU and 2GB memory to run all of our experiments unless otherwise noted.

7.2.1 DNS Overhead

IPA uses a signed TXT record in DNS to publish an Internet registry's revocation list (§ 4.2). An AS periodically downloads the revocation list from each registry. Each entry in a revocation list can be encoded in $\leq$30 bytes ($\leq$18 bytes for an IPv4 prefix in the dotted-decimal format, one byte for space, 10 bytes for the revocation time, and one byte for the line break). A publisher can compress a list (*e.g.*, using `gzip`) to reduce overhead. An AS also needs to download the list's signature ($\sim$300 bytes) and a few other DNSSEC records.

We assume that at any time, a registry at most revokes 1% of the total prefixes that it owns and does not reallocate them to others. We use `gzip` to compress each revocation list, and use `base64` to encode a compressed list so that it can be stored as a text record. The BGP report of February 2011 [15] shows that there are a total of

BGP Table Dump	
Date collected	08/01/2010
Number of ASes	35728
Number of IP prefixes	337K
BGP Update Trace	
Vantage point	route-view2.oregon-ix.net
Number of peers	37
Date collected	08/01/2010∼08/31/2010
Number of updates	118 million
Average arrival rate	44.1 updates/s

Table 1: **This table summarizes the BGP data we use in evaluating IPA's routing overhead.**

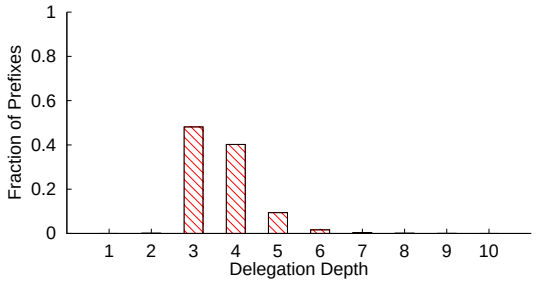

Figure 6: **The distribution of the depth of each prefix in the inferred IP prefix delegation hierarchy.**

37K ASes on the Internet. We assume that an AS downloads a revocation list once per day. This downloading frequency is acceptable, because it at most allows a prefix's previous owner to use the prefix for one extra day.

Figure 5 shows the average traffic load for serving the list at each Internet registry's DNS servers. As can be seen, even for the busiest registry ARIN, the estimated communication overhead is less than 10Kbps. This overhead is negligible compared to the regular load of a top-level DNS server, *e.g.*, the "M" root DNS server's regular load is over 32Mbps [10].

In the IPA design, an AS may also periodically download its certificate chains from the Internet registries to deal with key rollovers (§ 4.5). To evaluate this overhead, we assume that all ASes publish the IP prefix certificates they delegate to their children using DNSSEC. This places an upper bound on the top-level DNS servers' load. Each certificate includes three DNSSEC records and is about 650 bytes long (§ 4.1). We assume that each AS downloads its certificates once every day for each prefix it owns. Figure 5 shows the average traffic load from all registries for serving the certificate downloads. As can be seen, the IANA's DNS servers have the highest certificate serving overhead, but it is still much lower than a root DNS server's regular load, which suggests that IPA is unlikely to stress DNS.

7.2.2 Routing Overhead

We use trace-driven experiments to evaluate the overhead of IPA's in-band certificate distribution mechanism. We obtain a real BGP update trace from a RouteViews server [53]. Table 1 summarizes the BGP data we use. We then add IPA specific fields and updates to the trace to obtain a synthetic IPA BGP trace. We use the synthetic IPA trace to estimate the message overhead of distributing IP prefix certificates in-band. We also feed the IPA trace to a PC router running our IPA implementation, and measure the router's processing and memory overhead.

We generate the IPA BGP trace in three steps: 1) inferring IP prefix delegation hierarchy; 2) adding certificates for newly allocated and re-assigned prefixes; and 3) adding updates triggered by key rollover events. We describe each step in more detail.

First, we infer a prefix's delegation hierarchy to decide what certificates to add to a BGP update message announcing that prefix. We use a BGP table dump to infer this information. If an AS originates an IP prefix in the BGP table, we assume that it is the prefix's owner. If a prefix p' includes another prefix p, and both prefixes appear in the BGP table, we infer that p''s owner AS delegates the prefix p to p's owner. We also combine the IP prefix allocation records obtained from RIRs and IANA's websites to build the entire IP prefix delegation hierarchy. Figure 6 shows the distribution of the depth of the inferred hierarchy. More than 80% prefixes have a delegation depth of 3 or 4, suggesting that most ASes obtain IP prefixes directly from the RIRs or from provider ASes that directly obtain IP prefixes from the RIRs.

Second, we add prefix certificates to BGP updates that announce newly allocated or re-assigned IP prefixes. According to the IPA design (§ 3.3), an AS only sends an IP prefix certificate to a neighbor if it has not sent the certificate to the neighbor before. Thus, after the routing system has bootstrapped, only two types of updates carry IP prefix certificates: 1) an update that announces a newly allocated or re-assigned prefix, and 2) an update that carries new certificates generated during key rollovers (§ 4.5) for a previously announced prefix. We treat any IP prefix that has not appeared in the trace before as a newly allocated prefix, and any prefix whose origin AS has changed as a re-assigned prefix. To estimate the upper bound on the message overhead, we add the full certificate chain to each BGP update announcing a newly allocated or re-assigned prefix.

Finally, we add the update messages triggered by key rollover events to the IPA trace. Let a key rollover interval be T_r seconds. We let each AS randomly choose a key rollover time t during the T_r interval. We then add BGP updates that include the rekeying AS's new certificates for all its prefixes and its child ASes' prefixes at time t in our trace. We add updates for both routing and identity key rollovers (§ 4.5). We assume that as an upper bound, each AS changes its routing keys once a week,

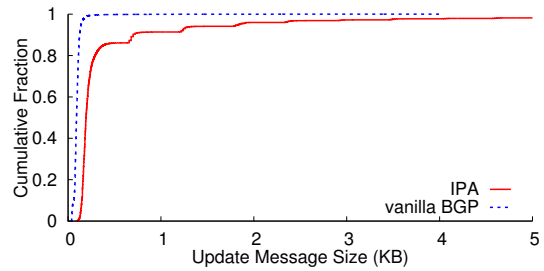

Figure 7: **The cumulative distribution of an IPA BGP update message size.**

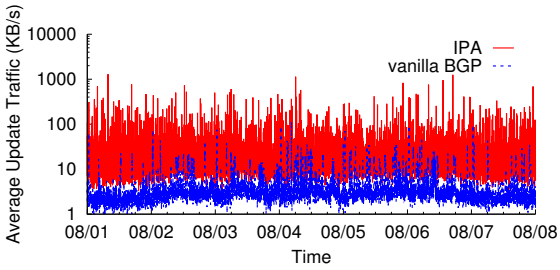

Figure 8: **The update traffic rate a RouteViews server sees averaged over 1-minute intervals during one week.**

and its identity keys once a month.

Message Overhead: Figure 7 shows the cumulative distribution of an IPA message size in one day's trace (August 1, 2010). The distributions in other days are similar and hence omitted. For comparison, we also show the distribution of an original BGP message size. As can be seen, over 80% of the IPA messages are smaller than 500 bytes. Given that each IP prefix certificate is around 650 bytes (§ 4.1), we can infer that over 80% of the messages do not carry any certificate, indicating that the caching mechanism described in § 4.3 is effective in reducing message overhead.

Figure 8 shows the IPA BGP update rate averaged over 1-minute bins in one week (August 1–7). The results during other weeks are similar and are omitted for clarity. For comparison, we also show the vanilla BGP update rate. The RouteViews server we use peers with 37 large ISPs. So we expect that the update process it sees is representative of what a BGP router sees in a large ISP [8]. The rate shown in Figure 8 is the aggregate arrival rate over all peers of the server. As can be seen, IPA increases the update traffic rate compared to the vanilla BGP. The 1-minute average aggregate update rate is usually less than 200KB/s. Since there are 37 peers, each peer on average receives less than 6KB/s update traffic. We think this overhead is acceptable compared to today's core routers' link capacities (10Gbps or 40Gbps).

Processing Overhead: We evaluate an IPA router's processing overhead by measuring 1) the fraction of CPU time it takes to process IPA's BGP messages, and 2) each

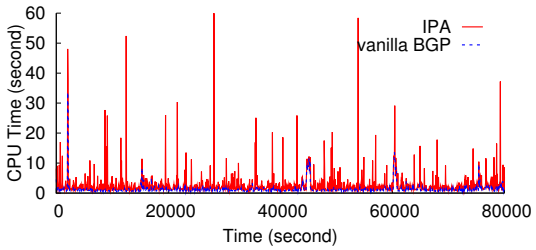

Figure 9: **The CPU time taken to process the messages received per 1-minute bin.**

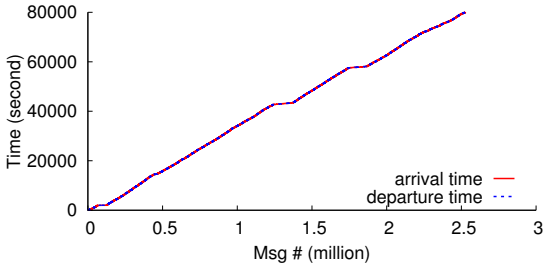

Figure 10: **The arrival and departure time of each message received during a day. The message number is in the unit of million (M).**

message's processing latency. We aggregate the BGP update messages into 1-minute bins to measure the CPU utilization. We feed the messages arrived in each bin to our IPA router implementation, measure the aggregate processing time, and compare it with the bin size.

Figure 9 shows the result during a one-day period (August 1, 2010) with 1-minute bins. The results for other days are similar and we omit them for clarity. For comparison, we also show the CPU time a XORP BGP router spends to process the original BGP trace. For each time bin, IPA takes more time to process the messages than the vanilla BGP, because it needs to validate new certificates piggybacked in the incoming messages. However, the CPU time that the router spends to process each 1-minute bin messages is usually less than 30 seconds, indicating that the router's CPU utilization is less than 50% and CPU is not a bottleneck. We may further improve our implementation's efficiency by applying instruction-level optimization to the RSA algorithm [40].

We further evaluate IPA's processing latency and examine whether it can keep up with the update arrival rate. We feed each update to the IPA router implementation according to the time it arrives. Figure 10 shows the arrival and departure time of each message. As can be seen, the arrival and departure lines almost overlap with each other, indicating that our implementation running on a commodity PC can keep up with the update arrival rate of the RouteViews server.

Memory Overhead: To evaluate IPA's memory over-

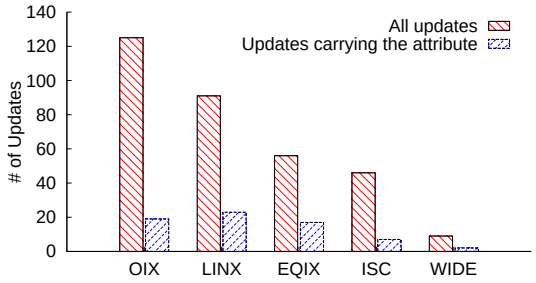

Figure 11: **The number of updates received by each RouteViews vantage point.**

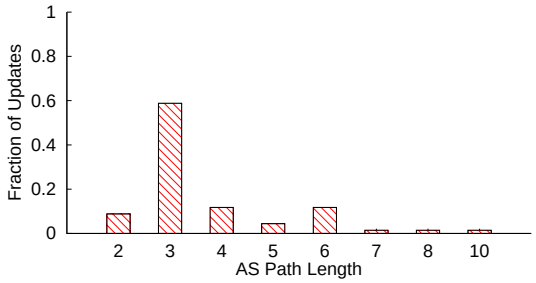

Figure 12: **The AS path length distribution of the received updates that carry the optional and transitive test attribute we inject. The path is from a RouteViews vantage point to the injection location.**

head, we feed the IPA BGP trace to our IPA implementation, and measure the memory needed to store all certificate caches. With our implementation, the trusted certificate cache consumes around 356MB memory using the BGP table data shown in Table 1. Our implementation stores only one physical copy for each certificate. The same certificates in different caches are pointers to the physical copy. The incoming cache uses ∼1.5MB memory to store the pointers. An outgoing cache uses at most 7MB, because it only need store a hash value for each certificate. This memory overhead is moderate because a router need not use these certificates in the packet forwarding time and can store them in low-cost DRAM.

7.3 Adoptability

In this section, we use real Internet experiments and analysis to evaluate IPA's adoptability. An adoptable design must satisfy two conditions: gradually deployable and providing incentives to early adopters.

7.3.1 Gradual Deployment

IPA uses the top-level DNSSEC infrastructure and BGP to certify and distribute IP prefix certificates. We evaluate whether early adopters can gradually deploy IPA in each system.

DNSSEC: First, we evaluate whether a legacy DNSSEC implementation can serve the DNSSEC records and revocation lists needed by IPA. We deploy a BIND9 DNS server which supports DNSSEC natively and has the largest installation base [16]. We use the DNSSEC signing tool bundled with the server software to generate the DNSSEC zone records for the IP prefixes allocated by IANA and all five regional Internet registries, and configure the server to serve the records and the revocation lists. We then use a legacy DNS client `dig` to fetch them. The `dig` client successfully retrieves all the records, indicating that the Internet registries can directly serve the DNSSEC records required by IPA without modifying DNS servers or breaking DNS clients.

BGP: We use BGP's transitive and optional path attributes to carry IPA-related fields. This design allows upgraded ASes to run the IPA protocols even if they are connected by legacy routers. This is because according to the BGP standard [51], legacy routers should forward any transitive and optional attribute.

To test IPA's compatibility with legacy BGP routers, we use a modified Quagga [11] BGP daemon to inject a BGP update with a transitive and optional attribute. We then monitor the propagation of this update from multiple RouteViews' vantage points. On August 27, 2010, we injected one such update to BGP using the BGP beacon platform maintained by RIPE RIS [13]. The update includes a previously unused prefix and a 3KB path attribute with an unknown type code 99. Figure 11 shows the number of updates observed by each RouteViews vantage point and among them how many still carry the attribute. For the updates still carrying the attribute, Figure 12 shows the AS path length distribution from their vantage points to the injection point. As can be seen, each vantage point observes at least one update carrying the attribute, and most of the updates carrying the attribute have successfully traversed multiple legacy ASes.

The RouteViews vantage points also receive many updates without the attribute. We suspect that this is caused by a Cisco software bug triggered by the injected update [17]. The bug causes certain Cisco router models to corrupt the path attribute. Consequently, a downstream router may reset the connection or remove the corrupted attribute. Given the prevalence of Cisco routers, we think that the result is encouraging. We expect that the affected routers will soon patch up this bug, and we will observe much more updates carrying the test attribute if we repeat this experiment.

7.3.2 Incentives for Early Adopters

We now discuss how the IPA design provides incentives for early adopters. Our analysis is based on the adoptability model presented in [26, 43]. The model assumes that each potential adopter is rational, and will have incentives to adopt a security mechanism if the security benefits outweigh the adoption costs. Because it is diffi-

cult to quantify costs, we use the model to qualitatively argue that IPA provides stronger incentives for adoption than previous work [34, 38, 46, 56, 57]. Thus, it is more likely to be adopted than previous work from a cost-effective perspective. We do not claim that IPA will be adopted, as many other factors (*e.g.*, politics) may affect the adoption process.

IPA's deployment involves four key parties: Internet registries, ASes, router vendors, and OS vendors. For simplicity, we focus on discussing the deployment incentives for the Internet registries and ASes, as past experiences of deploying DNSSEC [50] and IPv6 [36] suggest that they are often the deployment bottlenecks.

For the Internet registries, IPA achieves similar security benefits as previous work that requires a PKI [34, 38, 46, 56, 57], but has significantly lower deployment and management costs. This is because IPA uses the top-level DNSSEC infrastructure to bind an IP prefix to its owner's key. A DNSSEC-enabled registry need not deploy or manage any additional infrastructure to deploy IPA. Therefore, we believe that the Internet registries will have stronger incentives to deploy IPA than deploy a dedicated PKI required by previous work.

The IPA design also provides stronger deployment incentives for ASes than previous work, because ASes need not wait for the Internet registries to deploy a PKI and need not deploy additional certificate distribution infrastructures. Once the Internet registries have deployed IPA using DNSSEC, the top-level ASes that obtain IP prefixes directly from those registries can obtain immediate security benefits by distributing their IP prefix certificates in BGP and signing their BGP messages. These ASes will form a "club" to prevent prefix hijacking attacks within the club [26]. Using the IP prefix delegation hierarchy inferred in § 7.2.2, we find that such top-level ASes account for more than 78% of the total ASes. Once the top-level ASes have deployed IPA, their customers can obtain security benefits by adopting IPA, and so on. As the size of the protected club increases, the immediate security benefits that an adopter obtains also increase, which encourages more adopters, and can lead to a network effect of adoption [26].

7.4 Security Analysis

IPA bootstraps accountability with cryptography-based secure identifiers. Its security builds on the secrecy of private keys. The design stores private identity keys offline and uses periodic key rollovers to protect private keys. As long as the private keys remain secret, other security modules can use IPA to achieve accountable routing and forwarding, and DoS mitigation (§ 5).

The IPA design uses self-certifying AS identifiers. An AS may mint non-existent child AS identifiers by delegating sub-prefixes to those minted child ASes. However, because the minted identifiers are associated with sub-prefixes inside the AS's address space, the network can hold malicious ASes accountable by their address spaces to prevent them from evading traffic policing or gaining unfair shares of network resources (§ 5.3). An AS may inflate the AS path length in a BGP message by inserting the minted child AS identifiers, but it can achieve this goal by padding its own identifier in the message, which is a common BGP practice.

8 Related Work

The most related work in scope is the AIP architecture [20], which uses self-certifying identifiers as host addresses and domain identifiers. IPA retains the hierarchical IP addressing structure, but uses self-certifying AS identifiers. Unlike AIP, IPA's deployment does not require host re-numbering or trusted host hardware, but it requires the global root of trust of today's Internet (IANA) to continue to exist and function.

Public Key Infrastructures (PKIs) offer a hierarchical way to securely bind an identifier to a public key. Much existing work on secure routing, such as S-BGP [38], soBGP [57], psBGP [56], SPV [34], and Origin Authentication [46], requires the Internet registries to establish dedicated global PKIs to certify IP prefix ownerships or AS number ownerships. IPA obviates such requirements by using the existing top-level DNSSEC infrastructure to certify IP prefix allocations and using self-certifying identifiers as AS numbers. soBGP proposes to use a new type of BGP message to distribute various certificates in the routing system, while IPA uses a standard BGP extension to distribute IP prefix certificates.

The DNS CERT resource record (RR) [37] provides a generic way to store multiple types of certificates such as X.509, SPKI, and PGP with a DNS name. These certificates do not necessarily certify the DNS zone delegations, and hence do not certify IP prefix delegations. In contrast, IPA uses the Designated Signer and DNSKEY RRs rather than the CERT RR to map a reverse DNS zone delegation to an IP prefix delegation.

Simon et al. define network-layer accountability as traffic source identification and malicious traffic deterrence [54]. Their design assumes pairwise and transitive trust between ASes, and uses ingress filtering and an evil-bit in a packet header to stop DoS flooding traffic. However, if an AS within the trusted accountable group becomes compromised or malicious, it may fail to perform ingress filtering or set the evil-bit, rendering the design ineffective. IPA provides a similar form of accountability, but uses cryptography to establish accountability and is robust to malicious or compromised ASes.

An early version of IPA [58] outlines its main design modules. This work provides essential design details, an IPA prototype, and a comprehensive evaluation regarding

IPA's performance, adoptability, and security properties.

9 Conclusion

Lack of accountability makes the Internet vulnerable to many attacks, including source address spoofing, DoS flooding, prefix hijacking, and route forgery attacks. This work presents IPA, a design that bootstraps accountability in today's Internet with deployable and low-cost enhancements. IPA uses the top-level DNSSEC infrastructure to securely bind an IP prefix to an AS's public key and distributes these secure bindings using the routing system itself to lower deployment costs. We show that IPA enables a suite of security solutions [38, 43, 45] that collectively can combat the aforementioned network-layer attacks. We have presented the detailed IPA design, evaluated its performance, and shown that it is gradually deployable and provides stronger incentives for early adoption than previous proposals [34, 38, 46, 56, 57].

Acknowledgment

We thank Jeff Chase and the NSDI reviewers for their useful comments, and David Andersen for shepherding the paper. This work is supported in part by NSF awards CNS-0845858, CNS-1040043, and CNS-1017858.

References

[1] APNIC DNSSEC Service. http://www.apnic.net/services/services-apnic-provides/registration-services/dnssec.
[2] ARIN DNSSEC Deployment Plan. https://www.arin.net/resources/dnssec/index.html.
[3] BIND. https://www.isc.org/software/bind.
[4] DNSSEC Keys. http://www.ripe.net/dnssec-keys/index.html.
[5] DNSSEC Policy and Practice Statement. http://www.ripe.net/rs/reverse/dnssec/dps.html.
[6] DNSSEC Trust Anchors From ARIN. https://www.arin.net/resources/dnssec/trust_anchors.html.
[7] in-addr.arpa Transition. http://in-addr-transition.icann.org.
[8] Internet AS-level Topology on March 1st, 2011. http://irl.cs.ucla.edu/topology.
[9] IPv6 Support in BIND 9. http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch04.html.
[10] M Root DNS Server. http://m.root-servers.org.
[11] Quagga Routing Suite. http://www.quagga.net.
[12] RADb: Routing Assets Database. http://www.radb.net.
[13] RIS Routing Beacons. http://www.ripe.net/projects/ris/docs/beacon.html.
[14] SecSpider the DNSSEC Monitoring Project. http://secspider.cs.ucla.edu.
[15] CIDR Report. http://www.cidr-report.org, 2006.
[16] DNS Survey: October 2009. http://dns.measurement-factory.com/surveys/200910.html, 2009.
[17] Cisco Patches Bug That Crashed 1 Percent of Internet. http://www.reuters.com/article/idUS418825996320100831, 2010.
[18] DNSSEC Signatures in Reverse DNS Zones Now Enabled. http://www.apnic.net/publications/news/2010/dnssec-signatures, 2010.
[19] Root DNSSEC Status Update, 2010-07-16. http://www.root-dnssec.org/2010/07/16/status-update-2010-07-16, 2010.
[20] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable Internet Protocol (AIP). In *ACM SIGCOMM*, 2008.
[21] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, 2005.
[22] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, 2005.
[23] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, 2005.
[24] J. Bennett and H. Zhang. Hierarchical Packet Fair Queueing Algorithms. *IEEE/ACM ToN*, 5(5), 1997.
[25] M. A. Brown. Pakistan Hijacks YouTube. http://www.renesys.com/blog/2008/02/pakistan-hijacks-youtube-1.shtml, 2008.
[26] H. Chan, D. Dash, A. Perrig, and H. Zhang. Modeling Adoptability of Secure BGP Protocols. In *ACM SIGCOMM*, 2006.
[27] M. Crawford. Binary Labels in the Domain Name System. RFC 2673, 1999.
[28] Deterlab. http://www.deterlab.net.
[29] DNS Deployment Initiative. http://www.dnssec-deployment.org.
[30] H. Eidnes, G. de Groot, and P. Vixie. Classless IN-ADDR.ARPA Delegation. RFC 2317, 1998.
[31] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and Whitewashing in Peer-to-Peer Systems. *IEEE JSAC*, 24(5):1010–1019, 2006.
[32] A. Haeberlen, P. Kuznetsov, and P. Druschel. PeerReview: Practical Accountability for Distributed Systems. In *ACM Symposium on Operating Systems Principles*, 2007.
[33] M. Handley, E. Kohler, A. Ghosh, O. Hodson, and P. Radoslavov. Designing Extensible IP Router Software. In *USENIX/ACM NSDI*, 2005.
[34] Y. Hu, A. Perrig, and M. Sirbu. SPV: Secure Path Vector Routing for Securing BGP. In *ACM SIGCOMM*, 2004.
[35] Y.-C. Hu, D. McGrew, A. Perrig, B. Weis, and D. Wendlandt. (R)Evolutionary Bootstrapping of a Global PKI for Securing BGP. In *ACM HotNets-V*, 2006.
[36] G. Huston. Measuring IPv6 Deployment. http://www.internetac.org/wp-content/uploads/2010/02/apnic-v6-oecd1.pdf, 2009.
[37] S. Josefsson. Storing Certificates in the Domain Name System (DNS). RFC 4398, 2006.
[38] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (S-BGP). *IEEE JSAC*, 2000.
[39] O. Kolkman and R. Gieben. DNSSEC Operational Practices. RFC 4641, 2006.
[40] M. E. Kounavis, X. Kang, K. Grewal, M. Eszenyi, S. Gueron, and D. Durham. Encrypting the Internet. In *ACM SIGCOMM*, 2010.
[41] B. Lampson. Accountability and Freedom. http://research.microsoft.com/en-us/um/people/blampson/Slides/AccountabilityAndFreedomAbstract.htm, 2005.
[42] A. Li, X. Liu, and X. Yang. Dirty-Slate Accountable Internet Design. Technical Report 2010-07 (available at http://www.cs.duke.edu/nds/papers/ipa-tr.pdf), Duke University, 2010.
[43] X. Liu, A. Li, X. Yang, and D. Wetherall. Passport: Secure and Adoptable Source Authentication. In *USENIX/ACM NSDI*, 2008.
[44] X. Liu, X. Yang, and Y. Lu. To Filter or to Authorize: Network-Layer DoS Defense Against Multimillion-node Botnets. In *ACM SIGCOMM*, 2008.
[45] X. Liu, X. Yang, and Y. Xia. NetFence: Preventing Internet Denial of Service from Inside Out. In *ACM SIGCOMM*, 2010.
[46] P. McDaniel, W. Aiello, K. Butler, and J. Ioannidis. Origin Authentication in Interdomain Routing. *Computer Networks*, 50(16):2953–2980, 2006.
[47] P. Mockapetris. Domain Names – Concepts and Facilities. RFC 1034, 1987.
[48] J. Nazario. Estonian DDoS Attacks - A Summary to Date. http://asert.arbornetworks.com/2007/05/estonian-ddos-attacks-a-summary-to-date, 2007.
[49] J. Nazario. Georgia DDoS Attacks - A Quick Summary of Observations. http://asert.arbornetworks.com/2008/08/georgia-ddos-attacks-a-quick-summary-of-observations, 2008.
[50] E. Osterweil, M. Ryan, D. Massey, and L. Zhang. Quantifying the Operational Status of the DNSSEC Deployment. In *IMC*, 2008.
[51] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, 2006.
[52] P. Roberts. Massive Denial Of Service Attack Severs Myanmar From Internet. http://threatpost.com/en_us/blogs/massive-denial-service-attack-severs-myanmar-internet-110310, 2010.
[53] RouteViews Project. http://www.routeviews.org.
[54] D. R. Simon, S. Agarwal, and D. A. Maltz. AS-based Accountability as a Cost-effective DDoS Defense. In *USENIX HotBots*, 2007.
[55] Q. Vohra and E. Chen. BGP Support for Four-octet AS Number Space. RFC 4893, 2007.
[56] T. Wan, E. Kranakis, and P. van Oorschot. Pretty Secure BGP (psBGP). In *NDSS*, 2005.
[57] R. White. Securing BGP Through Secure Origin BGP. *The Internet Protocol Journal*, 2003.
[58] X. Yang and X. Liu. Internet Protocol Made Accountable. In *ACM HotNets-VIII*, 2009.
[59] X. Yang, D. Wetherall, and T. Anderson. A DoS-Limiting Network Architecture. In *ACM SIGCOMM*, 2005.
[60] A. R. Yumerefendi and J. S. Chase. Strong Accountability for Network Storage. *ACM Transactions on Storage*, 3(3), 2007.