Bootstrapping Accountability in the Internet We Have

Ang Li   Xin Liu   Xiaowei Yang
Dept. of Computer Science, Duke University

Abstract

The lack of accountability makes the Internet vulnerable to numerous attacks, including prefix hijacking, route forgery, source address spoofing, and DoS flooding attacks. This paper takes a "dirty-slate" approach to bring accountability to the present Internet with low-cost and deployable enhancements. Our design, IPA, uses the readily available top-level DNSSEC infrastructure and BGP to bootstrap accountability. We integrate it with a suite of security building blocks to combat various network-layer attacks. Our evaluation shows that IPA introduces modest overhead, is gradually deployable, and offers incentives for early adoption.

1 Introduction

Accountability, the ability to identify misbehaving entities and deter them, plays a critical role in achieving real-world security [35]. However, the Internet design has little built-in accountability: malicious hosts can send DoS flooding packets with spoofed source addresses to evade punishment; and malicious Autonomous Systems (ASes) can announce other ASes' IP prefixes, or assume other ASes' identities in the inter-domain routing system BGP.

The lack of accountability has led to many of the Internet's security vulnerabilities [18,47]. In this work, we ask the question: *can we overcome the Internet's security weaknesses with a minimal set of deployable patches?* That is, we aim to take a "dirty-slate" approach [22] to build an accountable Internet from the existing one.

This paper presents a dirty-slate design called IPA (IP made Accountable) that brings accountability to the Internet with only low-cost and gradually deployable enhancements. The IPA design faces several challenges. Chief among them is how to be both secure and lightweight. Accountability requires a secure binding between an entity's identity and its cryptographic keys so that a malicious entity cannot impersonate other legitimate entities or white-wash its tarnished identity. The present Internet uses two types of identifiers: IP addresses and AS numbers (ASNs), to identify network attachment points and ASes, respectively. Previous work [32,45,46] 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 [29], and web-of-trust lacks an authoritative trust chain to resolve conflicting IP prefix or ASN claims.

The IPA design uses two novel mechanisms to address this challenge. First, it leverages the top-level reverse DNSSEC hierarchy as a lightweight PKI to bind an IP prefix to its owner's public key (§ 3.2), securely certifying an IP prefix's ownership without a separate PKI. It uses the hash of an AS's public key as its self-certifying ASN, obviating the need for another PKI to certify ASN ownerships. We use DNSSEC [19–21, 41] because we can create a one-to-one mapping between an IP prefix delegation and a reverse DNS zone delegation, and 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 records as its owner's IP prefix certificate. Moreover, Internet registries are rapidly deploying the top-level reverse DNSSEC infrastructure [4,6,16,17]. The root and the `arpa` zone are already signed [17]. Deployment documents from major 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 that need to validate them to secure routing (§ 3.5). This design breaks 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 inband protocol both efficient and capable to support complex operations such as key revocations and rollovers.

The second challenge the IPA design addresses is how to bootstrap accountability in an adoptable manner, including being gradually deployable and incentivizing early adoption. We design IPA to be compliant with the existing protocols. 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 (§ 6.3.1). Different ASes can deploy IPA at different times without requiring 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 directly from the Internet registries can obtain immediate security benefits by preventing other ASes from hijacking their IP prefixes (§ 6.3.2).

We integrate IPA with several security building blocks [32, 37, 39] to prevent prefix hijacking, route forgery, source address spoofing, and DoS flooding attacks (§ 4). We show that IPA enables AS-level accountability, where other ASes can hold an AS accountable for the traffic it originates. Because an AS has incentives to protect legitimate hosts in its network, AS-level accountability motivates an AS to further implement host-level accountability, preventing an internal host from spoofing other internal hosts' addresses, and suppressing DoS flooding traffic originated from its network.

We have implemented IPA using XORP [27] and incorporated other security modules with it (§ 5). We evaluate IPA's performance and adoptability using trace-driven experiments (§ 6.2), live Internet experiments (§ 6.3.1), and analysis (§ 6.3.2). The results suggest that IPA is both efficient and adoptable 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 load. Its in-band certificate distribution protocol introduces modest overhead to a router. A single threaded IPA implementation running on a commodity PC can keep up the speed to process all messages arriving at a RouteViews server [43] that has 37 peers.

Our live Internet experiments show that IPA's protocol messages piggybacked in BGP can pass standard-compliant legacy routers. Our analysis suggests that IPA offers stronger incentives to early adopters than previous work that requires dedicated PKIs [32]. We show that once the top two levels of Internet registries have fully deployed reverse DNSSEC, more than 78% of the total ASes can immediately obtain their IP prefix certificates to prevent prefix hijacking attacks.

To the best of our knowledge, IPA is the first dirty-slate design that brings accountability to the present Internet in a secure, lightweight, and 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: The IPA design 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 we describe IPA's AS-level operations.

Trust Model: Similarly, the IPA design 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 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 identities. As we show in § 4, this ability further enables various security modules that can prevent prefix hijacking [28,32], route forgery [28,32], source address spoofing [37], and DoS flooding attacks [39].

Lightweight: We aim to introduce only lightweight enhancements to the Internet to meet our design goals. We hypothesize that enhancing the existing infrastructures with new functions has lower deployment costs than rolling out new global infrastructure services. For this reason, the IPA design does not require new global infrastructure services, unlike [10,32,46]; nor does it require trusted hardware at end systems (although it can help), unlike [18]. 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 legacy Internet components and ready to be deployed on today's Internet. IPA-enabled components should be able to communicate with each other even if there are legacy components between them.
- **Incentivizing Early Adoption:** IPA should provide immediate security benefits to early adopters to incentivize deployment. That is, an early adopter should not depend on many other entities to deploy IPA to gain security benefits.

The goal of being gradually deployable distinguishes a dirty-slate design from a clean-slate one. We believe that a dirty-slate approach can have a number of advantages. First, it is low risk and high reward. We can address the known weaknesses with the caution not to break what is working. Second, it can deliver benefits faster than a clean-slate one, because we need not build everything from scratch. Third, it can deepen our understanding on whether a clean-slate approach is inevitable. Only by examining the best possible dirty-slate design can we understand its limitations. Finally, even if a clean-slate approach is inevitable, a dirty-slate approach can offer temporary solutions to the present Internet's security problems during the transition period.

3 IPA Design

In this section, we describe how IPA instills accountability into the Internet. The design uses two key mechanisms to be lightweight and gradually deployable: 1) it leverages the top-level reverse DNSSEC hierarchy to bind an IP prefix to its owner's public key; and 2) it uses the existing BGP routing system to distribute IP prefix certificates in-band.

3.1 A Hybrid Approach to Secure Identifiers

The current Internet design 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 their IP prefixes to customers.

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. This design reduces deployment cost, as it obviates the need to bind an AS's identifier to its key using external trust anchors. This new ASN format can be gradually deployed in a similar manner as the recently deployed 32-bit ASN [44].

3.2 DNSSEC as a Lightweight PKI

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

There are several advantages of using DNSSEC to certify IP prefix delegation. 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 of trust (IANA). For example, when IANA delegates an IP prefix `165/8` to an RIR, it can certify this delegation by delegating the corresponding reverse DNS zone, `165.in-addr.arpa`, to the RIR. That is, it signs a Designated Signer (DS) record that includes the RIR's public key hash, and publishes the DNS entry. This delegation enables the RIR to further create a one-to-one mapping between the IP prefix and the reverse DNS zone's sub-delegations, *e.g.*, delegating `165.1/16` and `1.165.in-addr.arpa` to the same AS. This design reduces IPA's deployment cost at the Internet registries, as they need not maintain another infrastructure to certify IP prefix delegations. A prefix owner can use the DNSSEC records that certify its reverse DNS zone delegation as a certificate authorizing its prefix ownership (§ 3.2.1). We refer to this type of certificate as an IP prefix delegation certificate or a prefix certificate.

The second advantage is that Internet registries are rapidly deploying DNSSEC [26, 41]. The root zone was signed in July 2010 [17], and subsequently the `arpa` zone. The three largest RIRs, ARIN, RIPE, and APNIC, have all signed the reverse DNS zones for their address blocks using local trust anchors [4, 6, 16]. These zones account for 142 out of 175 sub-zones of `in-addr.arpa` [12]. At the time of writing, the only missing link is a signed `in-addr.arpa` zone, and ICANN (IANA's functional operator) is working on signing it [8]. Furthermore, the three largest RIRs 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].

Finally, because DNSSEC supports online queries, an Internet registry can use it to publish new IP prefix certificates to support key rollover (§ 3.6.2), or revocation (§ 3.4), 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 DNSSEC Records as IP Prefix Certificates

IPA uses three DNS resource record types associated with a reverse DNS name to encode an IP prefix certificate: the DS (Designated Signer) record, the public key (DNSKEY) record, and the signature (RRSIG) record of the DS record.

Figure 1 shows an example. When IANA allocates an IP prefix `165/8` to ARIN, it creates a DNSSEC entry for `165.in-addr.arpa` as the IP prefix `165/8`'s certificate. It uses the DS record to store the hash of ARIN's pubic key, and signs the DS record using its private key stored offline. It sets the inception and expiration times of the signature record (RRSIG) to the inception and expiration times of the IP prefix allocation. It then publishes the DNSSEC entry `165.in-addr.arpa` on its DNS servers. This process follows the standard DNSSEC practice, and also applies to IPv6 address allocation. Figure 2 shows the DNSSEC records that make up ARIN's IP prefix certificate for the prefix `165/8`.

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 allo-

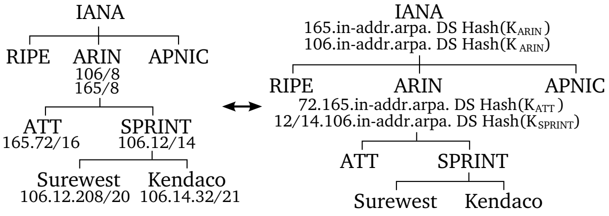
IANA
RIPE
ARIN
APNIC
106/8
165/8
ATT
SPRINT
165.72/16
106.12/14
Surewest
Kendaco
106.12.208/20
106.14.32/21
IANA
165.in-addr.arpa. DS Hash(K_ARIN)
106.in-addr.arpa. DS Hash(K_ARIN)
RIPE
ARIN
APNIC
72.165.in-addr.arpa. DS Hash(K_ATT)
12/14.106.in-addr.arpa. DS Hash(K_SPRINT)
ATT
SPRINT
Surewest
Kendaco

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

165.in-addr.arpa DNSKEY KARIN (290 bytes)
165.in-addr.arpa DS Hash(KARIN) (50 bytes)
165.in-addr.arpa RRSIG DS (312 bytes)

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

cate an IP prefix `106.12/14` to Sprint. We address this issue by extending the encoding format of a reverse 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. We omit them due to the lack of space, but include them in our technical report [36].

3.3 IP Prefix Sub-delegation

Once an AS obtains its IP prefixes, it may delegate subprefixes 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 rollover (§ 3.6.2). We believe that an AS has incentives to manage and publish its customers' certificates, because this 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.

3.4 IP Prefix Certificate Revocation

An IP prefix's parent owner may revoke a certificate 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 term of use or switches to a different ISP.

In the IPA design, a parent owner issues a new IP 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 old child owner's new key, or the parent's own key if it reclaims the IP prefix from a child owner. When multiple contiguous IP prefixes are de-allocated, a parent can aggregate these prefixes into a larger one, and issue a new certificate for the aggregated prefix to revoke previous delegations.

3.4.1 Revocation Notification

An AS participating in BGP needs to validate IP prefix certificates to secure routing (§ 3.5.3). To validate a certificate, an AS must check whether the certificate has been revoked or not. The IPA design uses a combination of push- and pull- based mechanisms to notify an AS of a certificate's revocation status.

Pushing New Certificates via Routing: As we will describe in § 3.5, IPA uses an in-band protocol to push prefix certificates to all ASes participating in BGP. If a new certificate's owner is an AS, the AS will use this push-based mechanism to notify other ASes of the old certificate's revocation by announcing the new one in BGP.

Periodic Pulling From Internet Registries: It is inconvenient for an Internet registry to announce new certificates and revoke the old ones using the push-based mechanism, because it may not participate in routing. We introduce a DNSSEC-based revocation list to address this issue. A revocation list includes the set of IP prefixes an Internet registry reclaims from its children, or re-assigns to children that are also Internet registries. The registry can publish the list using a TXT record of a special DNS name, *e.g.*, `revoked.arin.in-addr.arpa`, sign the list, and store the signature in a DNSSEC RRSIG record. An entry in a revocation list includes the revoked IP prefix and the revocation time. It revokes any certificate signed by the same registry that has an older inception time and certifies a prefix overlapping with the revoked prefix.

Each AS periodically (*e.g.*, on a daily basis) downloads the revocation lists from all Internet registries to invalidate revoked certificates (§ 3.5.3). An AS does not query DNS at the certificate validation time to reduce DNS servers' load. Periodic download may delay a certificate's revocation for a time period depending on the downloading frequency. But we consider this delay acceptable, because it will not lead to prefix hijacking attacks. Only the IP prefixes not allocated to any AS will suffer this delay, as an AS can immediately announce its new prefix certificate in BGP to declare its ownership.

3.5 In-band Certificate Distribution

ASes participating in routing need to obtain IP prefix certificates to validate IP prefix ownerships to secure routing (§ 4.1). IPA uses BGP itself to distribute certificates in-band. This design has two key advantages. First, it breaks the dependency loop between routing security and online certificate distribution, because it does not require a valid path between an AS and a certificate distribution server to exist before each AS obtains the necessary prefix certificates to establish a valid path. Second, it lowers deployment costs, as it does not need an out-of-band channel to deliver the certificates, unlike [32, 45].

The design of this in-band distribution protocol faces two key challenges: 1) Efficiency: how to keep the overhead low; 2) Correct validation: how to ensure that an AS can correctly validate an IP prefix's ownership. We describe how we address each challenge.

3.5.1 Full Chain of Trust for Correct Validation

In the IPA design, when an AS originates an IP prefix, it includes in its BGP message the chain of the latest (*i.e.*, not revoked) certificates associated with the IP prefix. We use a BGP feature, the transitive and optional path attribute, to carry them. 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 § 3.6.2.

Including the entire chain of certificates ensures correct validation (§ 3.7), because any AS receiving the BGP message can validate whether the origin AS owns the IP prefix by validating the chain of certificates, which further enables secure routing (§ 4.1). After an AS validates the certificates in a BGP message, it will further send them to its neighbors to which it announces the IP prefix in BGP messages.

3.5.2 Optimization to Reduce Overhead

The above design to ensure correct certificate validation incurs significant communication overhead, as it requires an AS to send a full chain of the certificates for each BGP message it sends. We use a simple but effective technique to reduce the overhead: each AS records the certificates it has sent to a neighbor and only sends to the neighbor the certificates that it has not sent yet.

An AS maintains several certificate caches to record what it has sent to a neighbor and to maintain certificate validation state. 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.

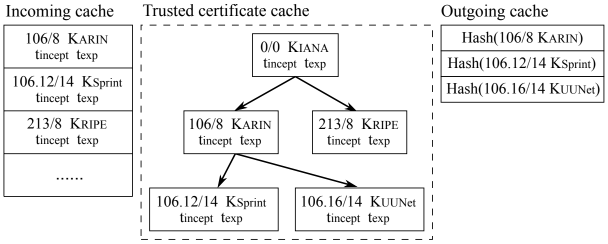

Figure 3: **An example of the certificate caches an AS maintains. It shows only one outgoing cache of the AS.**

When an AS receives an IP 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 regarding the IP prefix, it will first retrieve the full chain of certificates from its trusted certificate cache. It then examines the neighbor's outgoing certificate cache, and only sends the certificates that are not in the outgoing cache. Finally, it inserts the newly sent certificates into the neighbor's outgoing cache so that it will not send them to the neighbor again.

When an AS loses the peering connection to a neighbor, *e.g.*, due to 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.

3.5.3 Validating IP Prefix Certificates

An AS must validate a prefix certificate before it can use the certificate to secure routing (§ 4.1) or propagate it to its neighbors. Let C_{p_n} denote an IP prefix certificate that binds the prefix p_n to its owner's public key. The certificate C_{p_n} is valid if it meets the following conditions:

1. There exists a chain of certificates: $C_{p_0} C_{p_1} ... C_{p_n}$, such that: 1) C_{p_0} is the root IANA's self-signed certificate for IP prefix `0/0`; 2) every other certificate C_{p_i} is signed by its parent certificate $C_{p_{i-1}}$'s corresponding private key; and 3) each certificate's prefix p_i is a sub-prefix of its parent certificate's prefix p_{i-1}: $p_i \subseteq p_{i-1}$.
2. None of the certificates on the chain appears in an Internet registry's revocation list or is revoked by a newer certificate with an overlapping prefix.

To validate the certificates it receives, an AS organizes the certificates in its trusted cache in a tree-like structure, where a parent certificate's key signs the child certificates. Figure 3 shows an example of the trusted cache together with other certificate caches.

When an AS receives a list of certificates from a neighbor's BGP message, it first caches them in its incoming certificate cache. It then orders the certificates by their prefix length, and for each prefix p_{new}'s certificate C_{new}, starting from the shortest prefix, the AS validates C_{new} in the following steps:

Step 1: If C_{new} is signed by an Internet registry, the AS checks whether p_{new} overlaps with any prefix in the registry's revocation list. If it does and C_{new}'s inception time is older than the revocation time, the AS evicts C_{new}, *i.e.*, removing C_{new} from all its certificate caches.

Step 2: The AS then checks whether it has already cached C_{new} in its trusted cache. If so, it moves on to validate the next certificate. Otherwise, it looks for C_{new}'s parent certificate C_{parent} (identified by C_{new}'s signing key) in its trusted cache. If C_{parent} exists, and there does not exist any certificate $C_{conflict}$ among C_{parent}'s children whose prefix overlaps with that of C_{new} and that has a newer inception time, it validates C_{new} using C_{parent}'s public key. If the signature verifies, it inserts C_{new} into C_{parent}'s sub-tree in the trusted cache. If $C_{conflict}$ exists and is newer, it evicts C_{new}. If $C_{conflict}$ exists but is older, the AS instead evicts $C_{conflict}$, and removes the subtree rooted at $C_{conflict}$ from the trusted cache. However, the certificates in $C_{conflict}$'s subtree remain in the incoming cache, as they may be validated in the future (§ 3.6.3).

Step 3: If C_{parent} does not exist and C_{new} is the root certificate, the AS uses a hard-coded root key to validate C_{new}. Otherwise, the AS moves on to the next certificate, and marks C_{new} as unverifiable (cannot be validated or invalidated).

Step 4: If C_{new} is validated and inserted into the trusted cache in Step 2, the AS will look for any certificate in its incoming cache signed by C_{new}'s key, as it may become verifiable now. If such a certificate exists, the AS will repeat the above steps to validate it.

An AS should set a small upper bound (*e.g.*, < 10) on the number of unverifiable certificates it would receive from a neighbor, and disconnects the neighbor if the number exceeds the threshold. This is because a legitimate neighbor should only send validated certificates to it. Many unverifiable certificates from a neighbor signal malicious or faulty behavior.

3.6 Key Management

Like any cryptography-based system, IPA's accountability builds on the secrecy of private keys. We describe the preventive measures IPA takes to protect the secret keys: separating AS identity keys from routing signing keys and periodic key rollovers. We also discuss how to recover from a key compromise.

3.6.1 Separating Identity keys from Routing Keys

To secure routing, an AS must store its private key online to sign routing messages (§ 4.1). Yet it is desirable to keep a private key offline to reduce the risk of key compromise. To balance security and functionality, the IPA design separates an AS's identity key from its routing signing key. Recall that an IP prefix certificate we describe so far binds a prefix to an entity (an AS or an Internet registry)'s public key. The hash of the public key is also an AS's self-certifying identifier. We refer to this public/private key pair as an AS's identity keys.

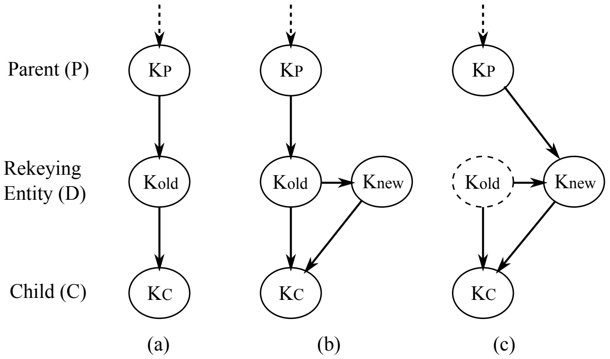

Figure 4: **This figure illustrates the key rollover process. Each node corresponds to a key, and an arrow points from a parent's signing key to a child's signed key. (a) shows the chain of trust before the key rollover happens. (b) shows the chains during the key rollover, where D signs a transient certificate to certify its new key K_{new} using its old key K_{old}. In (c), the key rollover process is over, and the old key K_{old} becomes invalid.**

To improve security, an AS generates a routing public/private key pair. For each IP prefix it owns, it will use its identity private key to sign a routing certificate that binds the IP prefix to its routing public 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, which can be validated using the algorithm described in § 3.5.3.

3.6.2 Key Rollover

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.

Identity Key Rollover: To improve security, it is also desirable to change an entity's identity keys periodically, but at a lower frequency than its routing keys, because the former is more involved. An entity must 1) request a new certificate from its parent; 2) revoke its old certificate; and 3) re-sign the child certificate with its new private key for each child to which it allocates a sub-prefix.

A key challenge we face is how to make a child certificate remain valid throughout a 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 so that both the child's old and new certificates are valid during a key rollover event. DNSSEC has a similar technique for key rollovers [33].

For clarity, we first describe the identity key rollover procedure for an AS. 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 its children 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 a publishing mechanism of its choice as described in § 3.3.

Each AS will periodically (*e.g.*, once a day) query its certificate issuers' publishing systems to download its latest chain 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 AS's certificate publishing system. Note that this online certificate download mechanism does not have a dependency loop on 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 there is a valid chain of trust reaching the certificate, including the link provided by the parent D's self-signed transient certificate. Figure 4(b) shows the new certificate chain.

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 sufficient long period d. The waiting period d should be long enough to ensure that each child AS of D has successfully downloaded and announced its new certificate chain in BGP. D can then announce its new certificates for its new key K_{new} in BGP to revoke its old certificates. 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.

If an Internet registry rekeys, the procedure is similar, except that the registry need not announce its new certificate in BGP, as its child ASes will obtain it via DNSSEC queries and announce it in BGP.

3.6.3 Recovering From Key Compromise

With the preventive measures we describe above, we expect key compromise to be a rare event in IPA. But for completeness, we describe how to recover from it.

Similar to a key rollover event, to recover from a key compromise, an entity must request each parent to certify the bindings between its IP prefixes and its new identity key, and use its new key to issue sub-prefix certificates for its children. Unlike a key rollover event, an entity must contact its parents and children offline to obtain its new certificates and to distribute the children's new certificates. This is because when its key is compromised, all its IP prefixes may be hijacked, disrupting its online communication.

Once the entity and its children obtain their new certificates, they should immediately announce them in BGP to recover their prefix ownerships. If an Internet registry's key is compromised, its new certificates can be announced by a child AS.

3.7 Property

We show that the IPA design has the following property:

Correct Validation: When an AS receives a BGP message announcing an IP prefix p, it must have also received a valid chain of certificates that certify the secure binding between p and its owner's public key.

This property enables an AS to correctly validate p's ownership (§ 3.5.3). We can prove this property using two levels of induction. First, we show that this is true when p's owner first announces it in the routing system. This is because with the IPA design, the owner will send p's full certificate chain to its neighbors (§ 3.5), and each of these neighbors will send any certificate on the chain that it has not sent to a neighbor to the neighbor, and so on. Second, we show that this remains true when any certificate on p's certificate chain is replaced by a new certificate. This is because p's owner will periodically download p's latest certificate chain, and announce any new certificate on the chain to its neighbors in BGP.

4 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.

4.1 Accountable Routing

IPA provides ASes with the necessary certificates to sign and validate routing messages. Hence, we can integrate it with a secure routing protocol such as S-BGP [32] to achieve origin authentication and AS path authentication.

Origin Authentication: An AS D that owns a prefix p can now sign its routing updates when it originates the prefix. Because of the correct validation property (§ 3.7), other ASes can use the chain of certificates that binds p to

D's public key to verify D's ownership of p, preventing malicious ASes from hijacking D's IP prefix 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 it to the next hop. A malicious AS cannot forge another AS's identifier because it cannot generate a valid signature. A transit AS can piggyback its public key in a BGP update message in the same manner as how prefix certificates are distributed (§ 3.5). One can also apply the same optimization technique as in § 3.5.2 to reduce the message overhead.

Although IPA uses self-certifying AS identifiers, it still effectively limits identity white-washing attacks where an AS abandons a tarnished identity and assumes a new one at will. This is because a valid AS identifier in IPA must be bound to an IP prefix via a chain of trust. To change its identity key, an AS must request a new prefix certificate from its parent; and changing the identity key more frequently than the regular key rollover frequency is a clear sign of anomaly.

4.2 Accountable Forwarding

The ability to securely sign BGP routing messages enables the deployment of Passport [37], a system that can achieve both packet source authentication and forwarding path inconsistency detection. This is because with IPA, an AS can piggyback a Diffie-Hellman public value in BGP messages that it originates and signs, and other ASes can verify that the value is indeed from this AS. This allows the Passport system to securely carry out a distributed Diffie-Hellman key exchange that establishes a shared secret between every pair of ASes.

Packet Source Authentication: In Passport, a source AS stamps a sequence of message authentication codes (MACs) into a packet header, using the secret key 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 policy. We can extend Passport to detect such behavior. This is because if a packet's forwarding path differs from the AS path its source AS selects to use, but its source address is authentic, an AS on the path will detect an invalid MAC, but the destination AS will detect a valid one. Thus, a destination AS can use this discrepancy to notify the source AS of the forwarding path inconsistency.

4.3 DoS Attack Mitigation

Finally, because IPA enables source authentication, it can readily integrate a DoS defense system that uses authentic source addresses to suppress attack traffic near its sources, *e.g.*, a filter-based system such as StopIt [38] or NetFence [39].

We extend NetFence and incorporate it with IPA, because NetFence scalably limits both denial of edge service (DoES) and denial of network service (DoNS) attacks without keeping per-flow state in the core routers. In DoES attacks, compromised nodes flood an innocent victim, while in DoNS attacks, compromised nodes collude into sender-receiver pairs to flood the network. NetFence introduces a secure congestion policing loop in the network to limit DoNS attacks. A NetFence packet carries unspoofable congestion policing feedback in a shim layer. An on-path router 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. It provides a legitimate sender its fair share of bandwidth regardless of malicious nodes' behavior. A receiver can use the unspoofable congestion feedback as network capabilities to suppress unwanted traffic, effectively limiting DoES attacks.

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. This motivates a source AS to prevent internal source address spoofing, and suppress DoS flooding traffic originated from its network. Differently, IPA uses hierarchical queuing [23] that follows the IP prefix delegation hierarchy to hold each AS accountable. That is, all traffic from the IP prefixes allocated to the same AS's public key will share one queue; a router may sub-divide 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 IP prefixes associated with the queue, not by the number of ASes sharing the IP prefixes. Hierarchical queuing prevents an AS from gaining unfair network resources by dividing its IP prefixes into many smaller ones and delegating them to minted identifiers.

5 Implementation

We have implemented a prototype of IPA's in-band certificate distribution mechanism (§ 3.5) using XORP [27], a Linux-based routing software suite. The implementation includes a standalone C++ library `libipa` that other implementations can incorporate. `libipa` implements certificate propagation and validation, and supports downloading revocation lists and importing new certificates for key rollovers from DNSSEC.

Our implementation addresses several practical issues that arise when an IPA router peers with a legacy router. First, we disabled the optimization technique (§ 3.5.2) 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 longer than 4KB message into smaller ones, each of which carries a subset of the certificates and public keys of the original message, and sends them in sequence to its legacy neighbor. The IPA router waits for a period of time longer than BGP's MRAI timer (*e.g.*, a few minutes) between sending out two consecutive messages, to avoid the first message being overwritten by the second one during propagation.

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

6 Evaluation

In this section, we evaluate IPA along four dimensions. First, we are curious to see whether the design works. Can the network bootstrap correctly without deadlocks? Will the key rollover procedure work without causing interruptions? 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 under various attacks.

6.1 Does it Work?

We run testbed experiments with our IPA implementation to gain insight on the consistency and correctness of the design. These experiments include: 1) bootstrapping experiments, 2) key rollover experiments, and 3) prefix hijacking experiments. We sample a small testbed topology from the AS-level Internet topology inferred from BGP table dumps. This topology includes six university ASes and all ASes on the shortest paths between them. It contains 17 ASes and 54 uni-directional links. 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 DNSSEC signing tool included in BIND9 [3] to generate the certificates. The topology includes four levels of IP prefix allocation: from IANA to an RIR, from an RIR to a top-level AS, and from the AS to a customer AS. We randomly pick three nodes as the root and two other RIRs' DNSSEC servers. We assume each AS's DNSSEC server is inside its network.

We then run the experiments on Deterlab [25]. Each node in the testbed corresponds to an AS. Each AS is configured with an initial IP prefix certificate chain. In a bootstrapping experiment, we observe that the system can successfully bootstrap as all certificates are validated and stored in each node's trusted cache. In a key rollover experiment, we observe that rekeying ASes can obtain IP prefixes from their parents and successfully propagate their new certificates, and each prefix always has at least one validated 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. The update message is rejected by all other ASes because there does not exist a certificate chain certifying the AS's ownership of the hijacked prefix.

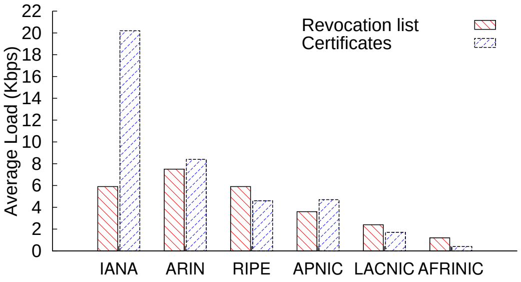

Figure 5: **The average communication overhead of each Internet registry to serve the revocation list and the IP prefix certificates. We assume all ASes use DNSSEC to publish the certificates they issue.**

6.2 Performance

To bootstrap accountability, IPA introduces overhead to both DNS and the routing system. We use analysis and trace-driven benchmarking experiments to evaluate this overhead, and show that IPA's overhead on DNS and the routing system is acceptable. We use a PC with Xeon 3GHz CPU and 2GB memory to run most of our experiments unless otherwise noted.

6.2.1 DNS Overhead

IPA uses a signed TXT record in DNS to publish an Internet registry's prefix revocation list (§ 3.4). An AS periodically downloads the revocation list. Each entry in a revocation list can be encoded in ≤30 bytes (≤18 bytes for a dotted-decimal format IPv4 address and its prefix length, 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 (around 300 bytes) and a few other DNSSEC records.

We use 1% of the total IP prefixes each registry allocates as the upper bound on the number of IP prefixes it revokes but does not re-assign at any time. We then use gzip to compress each revocation list, and use base64 to encode the compressed lists so that a server can store them as text records. The current BGP report [13] shows that there are a total of 35K ASes on the Internet. We as-

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∼08/27/2010
Number of updates	102 million
Average arrival rate	43.7 updates/s

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

sume each AS downloads a revocation list once per day. This downloading frequency is acceptable, because it at most allows the previous owner of an IP prefix to use the prefix for one extra day.

Figure 5 shows the average communication overhead for serving the list at each Internet registry's DNS servers. As can be seen, the overhead is low: even for the busiest registry ARIN, the estimated overhead is less than 10Kbps. Such 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 [7]).

In the IPA design, an AS may also periodically download its certificate chain from the Internet registries to handle key rollovers (§ 3.6.2). To evaluate this overhead, we assume all ASes publish the IP prefix certificates they issue using DNSSEC. This places an upper bound on the top-level DNS servers' load. Each certificate includes three DNSSEC records, which in total is about 650 bytes (§ 3.2.1). We assume each AS downloads its certificates once every day for each prefix it owns. Figure 5 shows the average communication overhead from all registries for serving the certificate downloads. As can be seen, the overhead of serving the certificates is higher than serving the revocation list for some registries, because an AS needs to download multiple certificates if it owns multiple prefixes, but only one copy of the revocation list. However, the highest load on IANA's DNS servers is still much smaller than the regular load of a root DNS server, which suggests that IPA is unlikely to stress DNS.

6.2.2 Routing Overhead

We use trace-driven experiments to evaluate the overhead of IPA's in-band certificate distribution protocol. We obtain a real BGP update trace from the RouteViews server [43]. We then adding IPA specific fields to each update message in the trace to generate the IPA BGP update trace. The IPA specific fields include IP prefix certificates and the public keys of the ASes on the path. We use the generated IPA trace to estimate the message overhead of distributing IP prefix certificates in-band. We also feed the IPA trace to a benchmark machine running our IPA implementation, and measure the machine's processing and memory overhead. Table 1 summarizes the data we use.

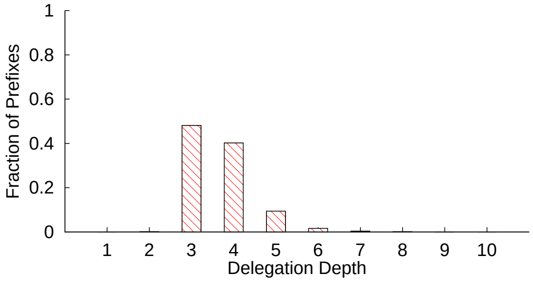

Figure 6: **The distribution of the depth of the inferred IP prefix delegation hierarchy.**

We estimate the number of IP prefix certificates on each prefix's certificate chain to generate the IPA trace. To do so, we first use the IP prefixes seen in a BGP table dump to infer the IP prefix delegation hierarchy. If we see an AS originates an IP prefix in the table dump, we assume it is the prefix's owner. If a prefix p' includes another prefix p, and both prefixes appear in the BGP table, we infer 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, which suggests that most ASes obtain IP prefixes directly from the RIRs or from provider ASes that in turn obtain address allocations from RIRs.

We further estimate which BGP update messages carry IP prefix certificates. According to the IPA design (§ 3.5), an AS only sends an IP prefix certificate to a neighbor if it has not sent the certificate to the neighbor before. Thus, only two types of routing updates need to carry IP prefix certificates: 1) an update that announces a newly allocated or re-assigned prefixes, and 2) an update that carries new certificates generated during key rollovers (§ 3.6.2) 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 upper bound the message overhead, we add the full certificate chain to each BGP update announcing a newly allocated or re-assigned prefix.

We add new updates to the trace to simulate key rollover events. Let a key rollover interval be T_r seconds. We let each AS randomly pick a key rollover time t during the T_r interval. We then add BGP updates that carry the rekeying AS's new certificates for all its prefixes and its child ASes' prefixes at the time t in our trace. We add updates for both routing and identity key rollovers (§ 3.6.2). We assume, as an upper bound, each

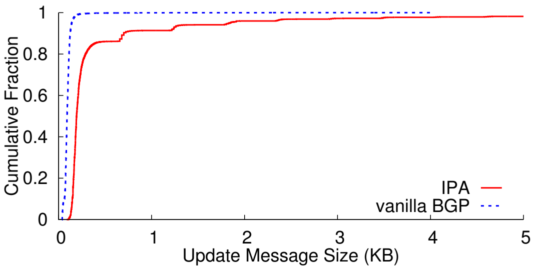

Figure 7: **The cumulative distribution of a IPA BGP update message size.**

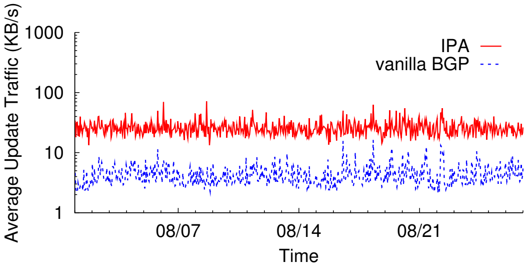

Figure 8: **The average update traffic rate the RouteView server sees during a 27-day period.**

AS changes its routing keys once a week, and its identity keys once a month.

Message Overhead: Figure 7 shows the cumulative distribution of the IPA message sizes in one day's trace (August 1, 2010). For comparison, we also show the distribution of the sizes of the original BGP messages. From the figure we can see, over 80% of the IPA messages are smaller than 500 bytes. Given that each IP prefix certificate is around 650 bytes (§ 3.2.1), we can infer that over 80% of the messages do not carry any certificate, suggesting that most messages do not announce newly allocated or re-assigned prefixes and are not triggered by key rollover events. This shows that our optimization mechanism described in § 3.5.2 is effective in reducing message overhead.

Figure 8 shows the simulated IPA BGP update rate averaged over an hour bin in a 27-day period. For comparison, we also simulate the vanilla update rate using the original BGP trace. The RouteViews server we choose peers with 37 large ISPs. So we expect the update process it sees to be representative of what a BGP router sees in a large ISP. As can be seen, IPA has increased the update traffic rate compared to vanilla BGP. The rate shown here is the aggregate arrival rate over all peers of the server. In most cases, the average aggregate update rate is below 100KB/s. Given that there are 37 peers, each peer on average receives less than 3KB/s. We think this overhead is unlikely to become a performance bottleneck compared to today's core link speed (10Gbps or 40Gbps).

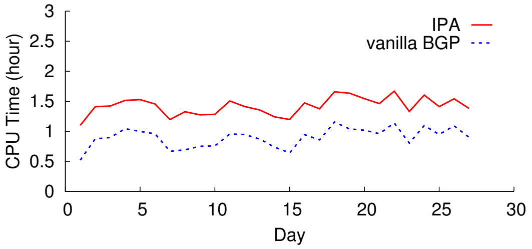

Figure 9: **The CPU time taken to process all messages received per day.**

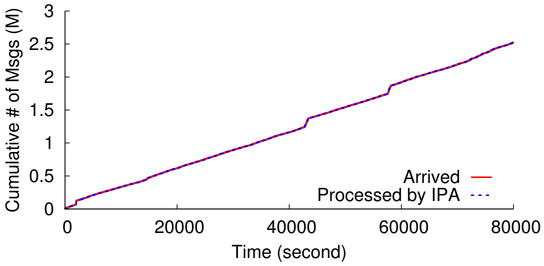

Figure 10: **The cumulative number of messages arrived and processed during a day. The number of messages is in the unit of million (M).**

Processing Overhead: We first evaluate the processing overhead of IPA by measuring the CPU time spent to process the update messages. To do this, we feed the IPA BGP update trace to our XORP implementation back-to-back, and measure the CPU time spent to process the messages received during each day. Figure 9 shows the result over a 27-day period. For comparison, we also show the CPU time spent by the vanilla BGP implementation processing the original BGP trace. As can be seen, for each day, IPA takes more time to process the messages that BGP, because it also needs to validate new certificates piggybacked in the incoming messages. Furthermore, the CPU time spent per day is less than 2 hours, indicating that our implementation is unlikely to stress the router.

We further evaluate IPA's processing delay and examine whether it can keep up with the update arrival rate. Figure 10 shows the cumulative number of messages arrived and processed during a one-day period (August 1, 2010). From the figure we can see, the two 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. The average processing delay observed by an update is only 0.19 seconds, negligible compared to BGP convergence time. The largest processing delay observed during the day's trace is 95.1 seconds. This is caused by a key rollover event of a large AS who owns over 3K IP prefixes. In our IPA trace, the updates triggered by a key rollover event are highly bursty, because they are added at the exact same time. In practice, because of the BGP pacing timers and different path latencies, the updates are likely to be more evenly distributed after the rollover event, which can help reduce the processing delay. We may further improve the efficiency of our implementation by applying instruction-level optimization on the RSA algorithm [34] and processing multiple messages in parallel.

Memory Overhead: To evaluate IPA's memory overhead, 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 of a certificate, and the same certificates in different caches are pointers to the physical copy. Therefore, the incoming cache only introduces about 1.5MB extra overhead to store the pointers. An outgoing cache costs at most 7MB, because it only needs to store a hash value for each certificate. This memory overhead is modest because the certificates are not used in the data plane and can be stored in low-cost SDRAM.

6.3 Adoptability

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

6.3.1 Gradual Deployment

IPA leverages the top-level DNSSEC infrastructure, and enhances the routing system to distribute IP prefix certificates in-band. 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 [14]. We use the DNSSEC signing tool bundled with the server software to generate the DNSSEC zone records for 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, suggesting that Internet registries can directly serve the DNSSEC records required by IPA without modifying DNSSEC servers or breaking DNS clients.

BGP: We embed all IPA specific information in BGP's transitive and optional path attributes. Upgraded ASes should be able to communicate with each other even if there are legacy ASes between them, because legacy routers, according to the BGP standard [42], should forward on any transitive and optional attribute.

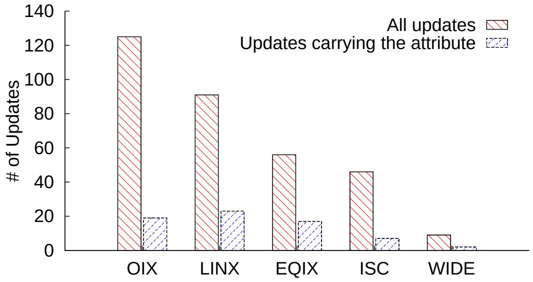

Figure 11: **The number of updates received by each RouteViews vantage point.**

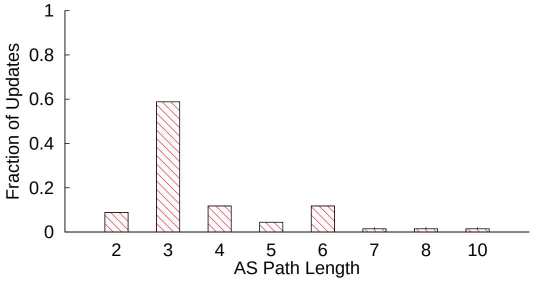

Figure 12: **The AS path length distribution of the received routes carrying the optional and transitive attribute. The path is from a RouteViews vantage point to the injection location.**

To test IPA's compatibility with legacy BGP, we use a modified Quagga [9] BGP daemon to inject a BGP update with a transitive and optional attribute. We then collect the BGP updates from multiple RouteViews' vantage points, and measure how many updates still carry the attribute. On August 27, 2010, we injected one of such updates to BGP using the BGP beacon system maintained by RIPE RIS [11]. 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 a vantage point to the update's injection point. From the figures, we can see that 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.

We note that there are also many updates received without the test attribute. We suspect this result might be due to a Cisco software bug triggered by the injected update [15]. The bug causes certain Cisco router models to corrupt the test attribute, and the downstream routers may reset the connection or remove the corrupted attribute. Given the prevalence of Cisco routers, we think the result is encouraging. We expect that the affected routers will have this bug patched up soon, and we will observe much more updates carrying the test attribute if we repeat this experiment again.

6.3.2 Incentives for Early Adopters

It is a challenging research topic by itself to systematically model real-world user incentives to adopt security enhancements. Hence, we do not aim to quantify user incentives to adopt IPA. Rather, we use a simple model to qualitatively argue that IPA provides stronger incentives for adoption than previous work that requires dedicated PKIs such as S-BGP.

We model a user's incentive (I) to deploy IPA as its immediate security benefits (F) minus its deployment cost (C): $F - C$. IPA's deployment involves four key user types: Internet registries, ASes (*i.e.*, ISPs), 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 [41] and IPv6 [30] suggest that they are often the deployment bottlenecks.

Compared to previous work which requires a dedicated PKI, the IPA design reduces an Internet registry's deployment cost, but achieves similar security benefits. This is because IPA leverages the top-level DNSSEC infrastructure to bind an IP prefix to its owner's key. An Internet registry has lower cost and thus stronger incentives: $I_{reg}^{\mathsf{IPA}} > I_{reg}^{SBGP}$. Let P_{reg}^{IPA} (respectively P_{reg}^{SBGP}) denote the likelihood that an Internet registry deploys IPA (SBGP). Stronger incentives imply a higher likelihood to deploy IPA than S-BGP: $P_{reg}^{\mathsf{IPA}} > P_{reg}^{SBGP}$.

An AS's security benefit of deploying IPA (or S-BGP) depends on whether the Internet registries and the provider ASes that are on its IP prefix delegation chain have deployed IPA (or S-BGP), because without their endorsements, other ASes cannot validate the AS's IP prefix ownerships. Because $P_{reg}^{\mathsf{IPA}} > P_{reg}^{SBGP}$, IPA provides an AS higher expected security benefit than S-BGP. Since an AS's cost to deploy IPA (for secure routing) is not higher than deploy S-BGP and can be lower if the AS has already deployed DNSSEC (§ 3.3), we conclude that an AS also has stronger incentives to deploy IPA than S-BGP: $I_{AS}^{\mathsf{IPA}} > I_{AS}^{SBGP}$.

Once the Internet registries have deployed IPA using DNSSEC, the top-level ASes that obtain IP prefixes from those registries can obtain immediate security benefits by distributing their IP prefix certificates and signing their prefix origin announcements to prevent malicious ASes from hijacking their prefixes. Using the IP prefix delegation hierarchy inferred in § 6.2.2, we find that such top-level ASes account for over 78% of the total ASes, suggesting the IPA design is amenable to ASes' early adoption. Once the top-level ASes have deployed IPA, their customers can obtain immediate security benefits by adopting IPA, and so on, leading to a network effect of adoption [24].

6.4 Security Analysis

IPA bootstraps accountability with cryptography-based provable identities, which build on the secrecy of private keys. So if an attacker compromises an Internet registry or an AS's key, it can impersonate the registry or the AS to disrupt routing and forwarding, including announcing the other entity's IP prefixes, or sending packets using its addresses. The IPA design stores private identity keys offline and uses periodic key rollovers to prevent private keys from being compromised. Compromised keys can be revoked using the mechanism described in § 3.4.

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 use hierarchical accountability to hold malicious entities accountable by the size of their address spaces (§ 4.3). Therefore, such an attack will not make the AS evade policing or gain unfair shares of network resources. 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, a common BGP practice.

7 Related Work

AIP [18] uses self-certifying identifiers as hosts' addresses and domain identifiers to bootstrap accountability. IPA retains the hierarchical IP addressing structure, but uses self-certifying AS identifiers. It bootstraps accountability by applying deployable patches to the Internet. IPA's deployment does not require host renumbering, but it relies on a global root of trust, which already exists in the present Internet.

Public Key Infrastructure (PKI) offers a hierarchical way to securely bind an identifier to a public key. Much existing work on secure routing, such as S-BGP [32], soBGP [46], psBGP [45], SPV [28], and Origin Authentication [40], requires the Internet registries to establish dedicated global PKIs to certify IP prefix ownerships and/or AS number ownerships. IPA obviates such requirements, by first leveraging the rapidly deploying top-level DNSSEC infrastructure to certify IP prefix allocation, and using self-certifying identifiers as AS numbers. This design reduces the deployment cost for the registries, and can incentivize adoption by offering immediate benefits to early adopters. 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. We have also optimized the distribution protocol, and evaluated its performance.

The DNS CERT resource record (RR) [31] provides a generic way to store multiple types of certificates, such as X.509, SPKI, and PGP, associated with a DNS name. But the key certified in a CERT record is not necessarily the Designated Signer for the DNS name. The IPA design uses the Designated Signer and DNSKEY RRs rather than the CERT RR to map a reverse DNS zone delegation to an IP prefix delegation.

An early version of IPA [47] outlines its main design modules. This paper provides many essential design details, including how to efficiently distribute prefix certificates in-band in BGP and how to perform key management. It also includes a prototype implementation and a comprehensive evaluation regarding IPA's performance, adoptability, and security properties.

8 Conclusion

The current Internet is vulnerable to a plethora of network-layer attacks, including source address spoofing, DoS flooding, prefix hijacking, and route forgery attacks. At the core of the security problems is the lack of accountability. This work presents IPA, a design that bootstraps accountability in today's Internet with deployable and low-cost enhancements. To be lightweight, IPA moves away from the traditional dedicated global PKIs. Instead, it uses the rapidly being deployed DNSSEC infrastructure to securely bind an IP prefix to an AS's public key. Furthermore, it distributes IP prefix certificates in the routing system itself, obviating the need for an out-of-band certificate delivery system. With the secure prefix-to-key bindings, IPA then enables a suite of security solutions that together can combat a wide spectrum of network-layer attacks. We have presented the detailed IPA design, evaluated its performance, and shown it is gradually deployable on the present Internet with strong benefits for early adoption.

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] M Root DNS Server. http://m.root-servers.org/.

[8] Progress on Signing the ARPA Tree. http://dnssec-deployment.org/pipermail/dnssec-deployment/2010-September/004360.html.

[9] Quagga Routing Suite. http://www.quagga.net.

[10] RADb: Routing Assets Database. http://www.radb.net.

[11] RIS Routing Beacons. http://www.ripe.net/projects/ris/docs/beacon.html.

[12] SecSpider the DNSSEC Monitoring Project. http://secspider.cs.ucla.edu.

[13] Cidr report. http://www.cidr-report.org, April 2006.

[14] DNS Survey: October 2009. http://dns.measurement-factory.com/surveys/200910.html, 2009.

[15] Cisco Patches Bug That Crashed 1 Percent of Internet. http://www.reuters.com/article/idUS418825996320100831, 2010.

[16] DNSSEC Signatures in Reverse DNS Zones Now Enabled. http://www.apnic.net/publications/news/2010/dnssec-signatures, 2010.

[17] Root DNSSEC Status Update, 2010-07-16. http://www.root-dnssec.org/2010/07/16/status-update-2010-07-16, 2010.

[18] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable Internet Protocol (AIP). In *ACM SIGCOMM*, 2008.

[19] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, 2005.

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

[21] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, 2005.

[22] H. Ballani, P. Francis, T. Cao, and J. Wang. Making routers last longer with viaggre. In *Proc. of USENIX Symposium on Networked Systems Design and Implementation*, Apr 2009.

[23] J. Bennett and H. Zhang. Hierarchical Packet Fair Queueing Algorithms. *IEEE/ACM ToN*, 5(5), 1997.

[24] H. Chan, D. Dash, A. Perrig, and H. Zhang. Modeling Adoptability of Secure BGP Protocols. In *ACM SIGCOMM*, 2006.

[25] Deterlab. http://www.deterlab.net/, 2010.

[26] DNS Deployment Initiative. http://www.dnssec-deployment.org/, 2009.

[27] M. Handley, E. Kohler, A. Ghosh, O. Hodson, and P. Radoslavov. Designing Extensible IP Router Software. In *USENIX/ACM NSDI*, 2005.

[28] Y. Hu, A. Perrig, and M. Sirbu. SPV: Secure Path Vector Routing for Securing BGP. In *ACM SIGCOMM*, 2004.

[29] 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.

[30] G. Huston. Measuring IPv6 Deployment. http://www.internetac.org/wp-content/uploads/2010/02/apnic-v6-oecd1.pdf.

[31] S. Josefsson. Storing Certificates in the Domain Name System (DNS). RFC 4398, 2006.

[32] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (S-BGP). *IEEE JSAC*, 2000.

[33] O. Kolkman and R. Gieben. DNSSEC Operational Practices. RFC 4641, 2006.

[34] M. E. Kounavis, X. Kang, K. Grewal, M. Eszenyi, S. Gueron, and D. Durham. Encrypting the Internet. In *ACM SIGCOMM*, 2010.

[35] B. Lampson. Accountability and Freedom. http://research.microsoft.com/en-us/um/people/blampson/Slides/AccountabilityAndFreedomAbstract.htm, 2005.

[36] A. Li, X. Liu, and X. Yang. Dirty-Slate Accountable Internet Design. Technical report, Duke University, 2010.

[37] X. Liu, A. Li, X. Yang, and D. Wetherall. Passport: Secure and Adoptable Source Authentication. In *USENIX/ACM NSDI*, 2008.

[38] X. Liu, X. Yang, and Y. Lu. To Filter or to Authorize: Network-Layer DoS Defense Against Multimillion-node Botnets. In *ACM SIGCOMM*, 2008.

[39] X. Liu, X. Yang, and Y. Xia. NetFence: Preventing Internet Denial of Service from Inside Out. In *ACM SIGCOMM*, 2010.

[40] P. McDaniel, W. Aiello, K. Butler, and J. Ioannidis. Origin Authentication in Interdomain Routing. *Computer Networks*, 50(16):2953–2980, 2006.

[41] E. Osterweil, M. Ryan, D. Massey, and L. Zhang. Quantifying the Operational Status of the DNSSEC Deployment. In *IMC*, 2008.

[42] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, 2006.

[43] RouteViews Project. http://www.routeviews.org/.

[44] Q. Vohra and E. Chen. BGP Support for Four-octet AS Number Space. RFC 4893, 2007.

[45] T. Wan, E. Kranakis, and P. van Oorschot. Pretty Secure BGP (psBGP). In *NDSS*, 2005.

[46] R. White. Securing BGP Through Secure Origin BGP. *The Internet Protocol Journal*, 2003.

[47] X. Yang and X. Liu. Internet Protocol Made Accountable. In *ACM HotNets-VIII*, 2009.