Internet Protocol Made Accountable

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

1. INTRODUCTION

The Internet design is vulnerable to numerous attacks, including source address spoofing, denial of service flooding, prefix hijacking, and route forgery attacks. Despite much research, this situation does not seem to improve: prefix hijacking or denial of service attacks continue to make headline news [10, 25]. This disheartening fact has prompted researchers to propose a radically different Internet architecture AIP [4] that replaces the aggregatable IP addresses with flat self-certifying addresses.

Although promising, the AIP design is not without challenges. Chief among them is deployability. AIP requires an overhaul of the Internet Protocol (IP). All hosts and networks must be re-numbered. Applications must be revised to use AIP addresses. Hosts also need special hardware "smart-NIC" to block DoS flooding traffic [4]. Every routing protocol, both intra-domain and inter-domain, must be revised to propagate AIP addresses, and routers must be upgraded to forward packets with AIP addresses. DNS must be extended to include AIP records, and so on. Moreover, AIP's flat addresses prohibit CIDR-style address aggregation, which is a best current practice for scalable routing [15].

In this paper, we ask the question: *can we design an Internet architecture that is as accountable as AIP but without its deployment and scalability tradeoffs?* To this end, we explore a design that provides accountability into the Internet while retaining the IP addressing structure. We refer to this design as IPa+ (standing for accountability enhanced Internet Protocol). The IPa+ design uses the chain of trust embedded in the Internet address allocation process to bind an address prefix to an authorized Autonomous System's (AS's) public key. Since AS numbers are flat identifiers that do not impact routing scalability, the IPa+ design uses the hash of an AS's public key as its self-certifying AS identifier in BGP. The IPa+ design uses DNSSEC [5, 7, 6, 23] to publish the secure bindings between an address prefix and its authorized AS's public key, as DNSSEC is being rapidly deployed by Internet registries [12]. A signed record in the reverse DNS zone (`in-addr.arpa`) serves as a lightweight certificate that secures a prefix-to-key binding. Routers may distribute these lightweight certificates as BGP attributes to secure routing. This design removes a significant deployment hurdle for securing BGP [17], as it obviates the need for an Internet registry to maintain an additional public key infrastructure.

The secure binding between a key and an IP address prefix bootstraps accountability in the network. It enables an AS to authenticate both its routing announcements and packets originated from its network. ASes can run a secure routing protocol (*e.g.*, sBGP [18]) to authenticate its routing announcements and prevents prefix hijacking and route forgery attacks. It can then use a source authentication system [20] that piggybacks a Diffie-Hellman key exchange in secure BGP announcements to allow ASes to share pair-wise secret keys. A source AS may use this key to authenticate packets originated from its network with low overhead [20]. Source authentication makes a sender accountable for its actions. It further enables simple DoS solutions that block attack traffic near its sources [22, 8, 30], and secure congestion policing mechanisms that prevent malicious flows from congesting the network to starve legitimate communications.

We evaluate the feasibility of IPa+ using data downloaded from regional Internet registries (RIRs) (§ 3). Our analysis shows that the load on the Internet registries is manageable, as the number of daily address prefix allocations at each RIR is small. We also present a preliminary comparison between IPa+ and AIP in terms of their security features, deployability, and scalability of IPa+ and AIP. Our study suggests that IPa+ provides nearly equivalent (if not stronger) security features to AIP and can be incrementally deployed on the IP network. It also requires fewer changes to the present Internet architecture, but some of the upgraded components (such as the access routers) in the IPa+ design implement more complicated functions.

2. IPa+ DESIGN

In this section, we describe the overall IPa+ design. Accountability in this paper refers to the ability to hold an entity responsible for its actions. The current Internet design lacks both host and network accountability. As a result, it is vulnerable to both data plane attacks and control plane attacks, including source address spoofing, DoS flooding, prefix hijacking, and route forgery. The IPa+ design aims to counter these attacks by introducing strong accountability into the IP-based network.

2.1 Bootstrapping Accountability

A key step to instill accountability is to use strong authentication to prevent impersonation attacks. One approach to enable strong authentication is to bind an entity's identifier to its public key through a trust anchor. Another approach is to use the hash of an entity's public key as its identifier without using a trusted anchor, *i.e.*, using self-certifying identi-

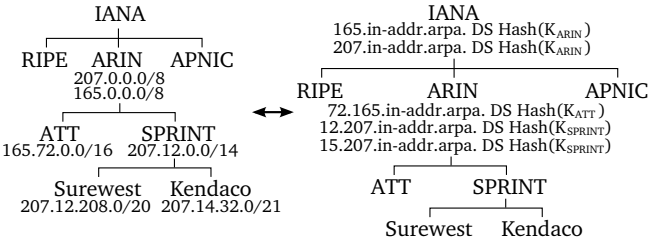

Figure 1: ***Left*: IP address allocation hierarchy; *Right*: the corresponding reverse DNS records that bind the prefixes to public keys.**

fiers. The AIP design takes the second approach, in which an end system's identifier or a network's identifier is its public key hash. Our design IPa+ takes a hybrid approach. We use trusted Internet registries to bind an IP address prefix to an authorized AS's public key, and use a hash of an AS's public key as the AS's identifier in BGP.

This design has several advantages. First, it facilitates deployment. Unlike AIP, hosts and networks need not change their addresses. Only an AS needs to change its AS number to its self-certifying identifier, but this change only effects the inter-domain routing protocol, which needs to be upgraded to a secure version anyway, regardless of whether we use an existing AS number (ASN) or a self-certifying AS identifier.

Second, the IPa+ design retains the aggregatable feature of IP addresses. Address aggregation enables the global routing tables to grow sub-linearly with the number of networks on the Internet [15], as smaller IP address prefixes can be aggregated into a larger one. Similarly, as one IP address prefix can summarize all hosts' addresses on the same subnet, routers need not announce individual host addresses in an intra-domain routing protocol. On the other hand, when scalability is not a concern, a large IP address prefix can be de-aggregated into smaller ones to facilitate traffic engineering. For instance, a multi-homed site may split its address prefixes into several smaller ones to balance the traffic from each of its providers.

Finally, we consider an assigned IP address more accountable than a self-certifying address. Although a self-certifying address prevents impersonation attacks, it permits white-washing attacks [13] where malicious nodes forgo their addresses and create new ones after they are caught misbehaving, *e.g.*, after they are blocked due to DoS flooding attacks. In contrast, an attacker could not white-wash its IP address by minting a new one as each address is assigned to it via external trust. We note that ASes in IPa+ cannot launch white washing attacks either, because a minted AS identifier that does not register with an Internet registry will not have a valid IP address prefix associated with it. Other ASes can detect its illegitimacy.

2.1.1 Key Bindings

We now describe how the IPa+ design securely binds an IP address prefix to an authorized AS's public key. This is done by following the chain of trust in the Internet address allocation process that involves two types of entities: Internet registries and ASes that participate in the Internet routing system. The root Internet registry IANA (Internet Assigned Numbers Authority) certifies the bindings between address prefixes allocated to RIRs and their public keys by signing digital certificates; and RIRs certify the bindings for address prefixes they allocate to ASes; and those ASes may in turn certify the bindings for address prefixes they allocate to their customers and so on.

One practical challenge is that Internet registries may be reluctant to deploy a public key infrastructure to issue and distribute digital certificates as it involves heavyweight operations such as setting up a Certificate Authority. Researchers have speculated that this challenge might have blocked the deployment of sBGP [17]. To address this challenge, we propose to use secure DNS (DNSSEC) to simplify the deployment hurdle, as DNSSEC is being rapidly deployed by Internet registries [12, 23]: by the end of 2009, ICANN will sign the root zone; and ARIN has already signed the reverse DNS zones of the address blocks allocated to it [2]. DNSSEC allows a zone owner to delegate a sub-zone to an entity by binding the sub-zone to the hash of its delegatee's public key, using a Designated Signer (DS) record. The delegator's signature together with an inception time and an expiration date will be stored in the corresponding RRSIG record.

The IPa+ design uses a signed reverse DNS record as a lightweight certificate to bind an address prefix to an authorized public key. In this design, the root registry IANA is the owner of the reverse DNS zone `in-addr.arpa`, as well as all IP addresses. When it allocates an address block, *e.g.*, 165/8 as shown in Figure 1, to an RIR (ARIN), it creates a DNSSEC record `165.in-addr.arpa`, stores the hash of the RIR's pubic key in a Designated Signer record, sets the expiration date of the record to be the expiration date of the address prefix allocation, and signs the record using its private key stored offline (following the standard DNSSEC practice). It then publishes the DNS record on its DNS servers. Similarly, when an RIR allocates an address prefix to an AS, *e.g.*, when ARIN allocates 165.72/16 to AT&T, it signs and publishes a reverse DNSSEC record `72.165.in-addr.arpa`.

A 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 address prefix 207.12/14 to Sprint. The reverse DNS zone of this address space includes four domains: from `12.207.in-addr.arpa` to `15.207.in-addr.arpa`. Signing and storing all those domain records may increase a DNS server's load and zone file size. For instance, a /9 prefix includes 128 DNS domains. IPa+ uses a security feature of DNSSEC: the Next Secure (NSEC) record to address this issue. An NSEC record is a secure way for a DNS server to return an NXDOMAIN (domain not existent) answer. As a DNSSEC server cannot presign NXDOMAIN responses for all non-existent domains, it uses a NSEC record to link all existent domain names. A DNSSEC server sorts registered domain names in its zone, and for each domain name record, it adds a NSEC record that points to the next existent domain name.

With the help of NSEC records, we can use two domain names: one for the left boundary of an address prefix, and the other for the right boundary, to publish an address prefix allocation result. In both records, the DS record is set to the address assignee's public key hash. For instance, in the example of 207.12/14, a DNS record for `12.207.in-addr.arpa` and one for `15.207.in-addr.arpa` are created. Any query for an address prefix within the range 207.12/14 would return the DNSSEC record for `12.207.in-addr.arpa`, with an NSEC record pointing to `15.207.in-addr.arpa`. If both records have the same public key hash in their DS records, it certifies that the address prefix 207.12/14 are allocated to the public key.

2.1.2 Address Prefix Sub-allocation

An AS x that directly obtains an address prefix p_x from an Internet registry may sub-allocate a smaller address prefix $p_c \in p_x$ to a customer c. For instance, in Figure 1, Sprint allocates a sub-prefix 207.12.208.0/20 to Surewest from its address prefix 207.12/14. In IPa+, an AS signs an address delegation attestation to certify this allocation. An address delegation attestation includes the inception time of the address delegation and its expiration time. The former is used as a sequence number for an AS to revoke an old delegation.

We note that an AS may store and publish an address sub-allocation result using DNSSEC, as does an Internet registry. But IPa+ does not mandate an AS to use DNSSEC to reduce the routing system's dependency on DNS. The customer to which an address prefix is allocated will use BGP to disseminate the allocation result to all ASes participating in the routing system (§ 2.2).

2.1.3 Revoking a Key Binding

We consider two cases for key revocation. In the first case, an address prefix is assigned to a new key, so the old prefix-to-key binding must be revoked. This case may occur when an AS's old key is compromised, or when the address prefix is allocated to a different AS. In the second case, a prefix-to-key binding is no longer valid, for instance, an Internet registry may terminate an AS's address allocation because the AS violates its terms of agreement, but the address prefix is not allocated to a new key.

We handle these two cases separately. If the address prefix is assigned by an Internet registry, in the first case, the registry must revoke the compromised key by publishing a new DNSSEC record with the DS field set to the authorized new key; in the second case, the Internet registry will publish the revoked address prefix in a revocation list via a DNSSEC TXT record of a special DNS name, *e.g.*, `revoked.arin.in-addr.arpa`, and signed by the registry's privacy key. Similarly, if the address prefix is assigned by an AS to a customer, in the first case, the AS revokes the compromised key by issuing a new address delegation attestation; in the second case, the AS will sign a new self-delegation attestation using its private key, re-allocating the address prefix to itself.

2.2 Securing Routing

Once an address prefix is securely bound to an AS's public key, the AS could use mechanisms similar to sBGP [18] to secure its routing announcements. An AS that announces an address prefix may announce the chain of signed DNS records and address delegation attestations using BGP attributes periodically. As IPa+ uses an AS's public key hash as its identifier, an AS's public key need not be certified. An AS may distribute its public key via another BGP attribute. An AS signs its routing announcements as in sBGP to prevent prefix hijacking or route forgery attacks.

To validate a routing announcement, an AS is hardcoded with the root Internet registry IANA's public key. An AS can use this key to validate IANA's delegation to an RIR's public key , and use the RIR's public key to validate its delegation to an AS, and so on.

An AS must ensure that a prefix-to-key binding is valid before it uses the key to valid any prefix sub-allocation. An AS can do so by querying DNS in real-time, but it increases the load on DNS. To be scalable, we design IPa+ to use new bindings to automatically revoke old bindings, and to periodically check DNS at a low frequency to detect revoked address prefixes that are not allocated to new keys yet. That is, when an AS's prefix-to-key binding is revoked but a new binding is available, the AS will immediately announce the new binding into the routing system. Other ASes will use this new binding to invoke any old binding, as the new binding must have a more recent inception time. Each AS also periodically queries RIRs' DNS servers to obtain and cache the revocation lists, and uses the list to detect any stale binding. This checking is not urgent, because the address prefixes in those revocation lists are not allocated to any other ASes, and a stale binding at most allows an AS to use the address prefix for a longer period, but does not allow it to hijack other ASes' address prefixes.

2.3 Authenticating Packet Origins

Secure routing further enables an AS to authenticate packets originated from its network using a source authentication system Passport [20]. In Passport, ASes piggyback a Diffie-Hellman key exchange in their BGP announcements. That is, each AS i generates a Diffie-Hellman public/private value pair (b_i, r_i) and inserts the Diffie-Hellman public value in its BGP announcements. Using a standard Diffie-Hellman construct, every other AS j combines its private value r_j with AS i's public value b_i to obtain a shared pair-wise secret key: $K(i,j) = b_j^{r_i} \ mod \ p = b_i^{r_j} \ mod \ p$, where p is a system-wide parameter. With secure routing, this Diffie-Hellman key exchange is secured, as an AS signs its routing announcements. Thus, no attacker can tamper its Diffie-Hellman public value.

An AS authenticates a packet originated from its network by computing a secure token for each AS on the packet's path with the key it shares with the AS. It inserts these tokens in a shim layer between IP and an upper layer protocol in the packet. The destination AS and an intermediate AS on the

path can use the secret key it shares with the source AS to validate that the packet is indeed from the source AS, as it is computationally infeasible for an AS that does not know the shared key to compute a valid token.

Passport allows each AS to use a local mechanism to prevent hosts in its network from spoofing each other's addresses, as routers in an AS are under the same administrative domain and can trust each other. An AS that fails to do so may harm hosts within its network, but has little impact on other networks, as it cannot spoof other ASes' address prefixes, and the network can isolate traffic originated from the AS based on their address prefixes in face of attacks [22]. An AS may choose a scheme such as ingress filtering [14], SAVE [19], or a secure MAC layer protocol [1] that best suits its network to prevent internal spoofing.

2.4 Stopping Unwanted Traffic

Once the network can identify a packet's origin, it may use a DoS solution [22, 8, 30, 24] to stop unwanted attack traffic. For instance, one of those solutions, StopIt [22], enables a receiver to send filter requests to block unwanted traffic. As the origins of the unwanted traffic cannot be spoofed, the network can direct the filter requests to ASes that source the attack traffic, and places filters right at the access routers of the attack sources.

2.5 Scalably Limiting Malicious Flows

Another form of DoS attacks involves malicious nodes organizing into sender/receiver pairs to flood a link and starve legitimate communications. This type of attack does not involve unwanted traffic so that DoS solutions [30,29,22,8,24] designed to give a receiver the control to stop unwanted traffic become ineffective.

The IPa+ design is able to mitigate this type of attack with authenticated source addresses. A simple but less scalable solution is to use fair queuing [11, 26] to prevent malicious flows from starving legitimate ones. However, this solution requires per-sender or per-flow queues, and may not scale well if there are many (*e.g.*, a few million of) attackers.

The IPa+ design uses the pairwise secret keys obtained from Passport to develop a solution (NetFence) that enables the network to scalably police malicious flows. NetFence aims to enforce all senders, including misbehaving ones, to follow congestion policing policies without keeping per-sender (or per-flow) state. At a high level, NetFence works as follows. A congested AS on a packet's path uses the secret key it shares with the source AS of the packet to place a secure congestion feedback in a packet header. This feedback cannot be tampered by malicious nodes and will be returned to a sender by a receiver. A sender's access router checks the returned congestion feedback, and enforces a TCP-like behavior on each sender. That is, if the downstream AS is congested, the sender must reduce its sending rate multiplicatively; otherwise, it may increase its sending rate additively. More details about NetFence can be found in [21].

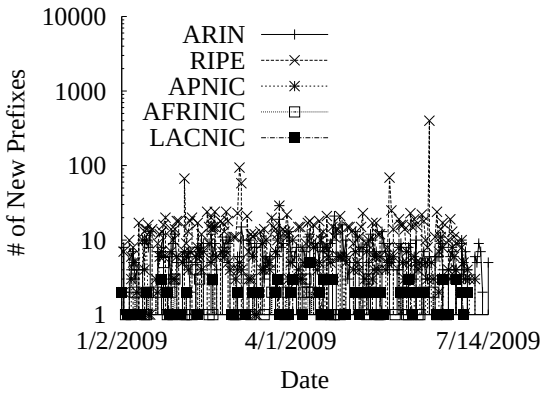

Figure 2: This figure shows the daily new address prefix allocated by RIRs from January 2, 2009 to July 14, 2009.

3. PRELIMINARY EVALUATION

In this section, we present a preliminary evaluation of IPa+. We are interested in estimating the load on Internet registries to sign DNSSEC records that bind address prefixes to authorized public keys, and the load on DNS servers to distribute the revocation lists. This is because these secure bindings bootstrap accountability, and whether Internet registries or DNS servers can handle the load impacts the feasibility of IPa+. We also compare IPa+ with AIP to evaluate its security, deployability, and scalability.

3.1 Signing Overhead for RIRs

In IPa+, an RIR signs at most two Designated Signer (DS) records for each prefix allocated to an AS. To estimate the signing overhead at an RIR, we download the daily summaries of the RIRs' allocated addresses from ARIN's FTP server [3], and compute the daily new prefix allocation in 2009. ARIN's server mirrors the address allocation results from the other four RIRs: RIPE, APNIC, AFRINIC, and LACNIC. Figure 2 shows how many new prefixes are allocated by each RIR every day from January 2, 2009 to July 14, 2009. We can see that during this period of time, the number of new prefixes allocated every day is small, typically around $30 \sim 50$ for all RIRs. There are a few spikes in the RIPE data set with a daily maximum of less than 400 prefixes. We have manually inspected these spikes and found that they are often caused by data re-organization at RIPE due to historic reasons. For instance, the highest spike shows that there were 400 new prefixes added to RIPE's address allocation list on June 27. But we find that 388 of them were allocated a long time ago, and some of them preceded when RIPE was established. This suggests that the DNS record signing load at Internet registries would be low.

3.2 DNS Query Overhead for RIRs

In the IPa+ design, an AS needs to periodically query RIRs' DNS servers for the set of RIR revoked address prefixes that are not allocated to other ASes (§ 2.2). We estimate the query load on DNS as follows. As described in § 2.1.3, a revoca-

RIR	Number of Prefixes
ARIN	50877
RIPE	42968
APNIC	31928
AFRINIC	2088
LACNIC	12922

Figure 3: Number of prefixes directly allocated by RIRs.

tion list is published as a DNS TXT record. An entry in a revocation list would include an address prefix and the inception time of the revocation, and can be encoded with $\leq$30 characters: $\leq$18 characters for a dotted-decimal format IPv4 address and its prefix length, one character for space, 10 characters for the inception time, and one character for line break. The signature field for an DNS record is about 2096 bytes.

Figure 3 shows the number of address prefixes allocated by each RIR as obtained from their whois summaries. If we assume the maximum number of RIR revoked but not reallocated address prefixes is 1% of the total address prefix it allocates, then the largest revocation list at ARIN is about 17K bytes. If each AS downloads this list once a day, and since there are less than 35K ASes on the Internet, the query traffic on an RIR's DNS server would be less than 60Kbps.

3.3 Comparing with AIP

We compare IPa+ with AIP from three dimensions: security, deployability, and scalability. We omit the comparison results for scalability, as IPa+ has the same scaling factor as IP, and it has been shown in [4] that AIP's BGP table sizes could be 5$\sim$9$\times$ larger than those on the Internet. The difference in terms of intra-domain routing table size could be even larger, as AIP needs to keep an entry for each host's identifier.

3.3.1 Security

Secure binding between an address and a public key: Both AIP and IPa+ provide this feature. IPa+ uses the existing Internet registries as trust anchors to create a binding, while AIP uses self-certifying identifiers and a global registry to create the binding. A host's AIP address consists of both an accountability domain ID (AD) and an end system ID (EID): AD:EID. The binding between EID and a host's public K_{EID} is self-certifying, but a host and its domain AD must sign the address AD:EID and publish it in the global registry to bind the end system's key K_{EID} to its full address AD:EID and to revoke the binding if a key is compromised. IPa+ uses the chain of trust in the existing Internet address allocation process to bind an authorized AS's public key to an address prefix and to revoke the binding if the key is compromised. IPa+'s binding is more coarse grained as a key is not bound to an individual IP address.

Source accountability: Both IPa+ and AIP are able to prevent source address spoofing attacks. Differently, IPa+ places authentication tokens in packet headers, while AIP uses a combination of unicast reverse path forwarding (uRPF) [9] and on-demand challenge-and-response to validate a packet's source address.

With AIP, if a packet's AD address passes a router's uRPF filters, the router considers its source address valid; otherwise, the router sends a challenge to the sender, and the sender must sign the challenge with its private key to validate its EID. Routers cache the mapping between validated source addresses and their incoming interfaces. Compromised ADs on the path can send packets with spoofed AIP addresses as well as minting new EIDs. For ADs that pass uRPF filters, they may spoof EIDs residing in other ADs. With Passport, a malicious AS cannot spoof any address within other ASes' address space but may send packets with any address within its own address space, even if that address does not correspond to a physical host. A compromised AS may duplicate packets it forwards, but unlike AIP, it cannot spoof arbitrary packets of those ASes' addresses for which it provides forwarding service.

Routing attack prevention: Both IPa+ and AIP can prevent prefix hijacking and route forgery attacks using mechanisms similar to sBGP [18].

DoS attack prevention: Both IPa+ and AIP enable a DoS victim to stop unwanted traffic. IPa+ places trust in the network and relies on routers to stop unwanted traffic, while AIP places trust in host hardware, and relies on hosts to install a smart network interface card to stop unwanted traffic.

Malicious flow mitigation: The IPa+ design uses a robust congestion policing protocol to prevent malicious flows from flooding a link and enforce TCP-like congestion avoidance behavior. AIP does not prevent this type of attack.

3.3.2 Deployability

It is a challenging task to compare the deployability of two network architectures. We defer a comprehensive study on this topic to future work, but use a simple metric: the components that need to upgrade and the type of upgrade to compare AIP and IPa+. We reckon that this simple binary metric (upgrade or not) does not capture the complexity of the type of upgrade, but we use it as a first-order approximation to gain insight, because intuitively, fewer upgraded components implies less dependency, which facilitates deployment.

Figure 4 summarizes the comparison result. IPa+ requires fewer modifications to the existing Internet architecture, mainly because it is built on IP. Routers (other than access routers or AS border routers) need upgrade only when they are bottleneck routers to benefit from IPa+'s robust congestion policing mechanism. Note that although IPa+ uses secure DNS to distribute key bindings, it does not modify the protocol, while AIP requires protocol modifications.

There are two other important aspects that can measure deployability: backward compatibility and deployment benefits for early adopters. IPa+ is backward compatible and its source accountability scheme provides incentives for early adopters [20]. We leave the comparison on these two aspects for future work, as the AIP design does not specify an incremental deployment plan.

	Host OS	Host HW	Applications	Access Router	Border Router	Other Routers	secure DNS	Intra-AS Routing	BGP	Registries
AIP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓, for key registration and revocation only
IPa+	✓	×	×	✓	✓	×, if non-bottleneck; ✓, otherwise	×	×	✓	✓, for address allocation, key registration, and revocation

Figure 4: The comparison between components that require upgrade in AIP and IPa+. ✓ indicates upgrade required; × indicates not required.

4. OTHER RELATED WORK

Similar to several secure BGP proposals [18, 28, 27, 16], IPa+ uses the chain of trust embedded in the Internet address allocation process to bind an address prefix to an authorized public key, but it uses self-certifying AS numbers to prevent route forgery, and DNSSEC rather than a separate PKI to certify a key binding. SBGP requires Internet registries to deploy public key infrastructures such as X.509 to bind both an address prefix and an ASN to a public key [18]. SoBGP uses web-of-trust to bind an address prefix and an ASN to a public key, but does not specify how to validate a web-of-trust certificate [28]. PsBGP binds an ASN to an authorized public key via Internet registries, but trusts peering ASes to certify the address prefixes allocated to each other [27]. SPV [16] only binds an address prefix to an authorized public key, but allows malicious ASes to insert other ASNs in BGP announcements to forge routes.

There is a plethora of work on preventing source address spoofing attacks [14, 9, 20, 19, 4]. The IPa+ design uses Passport [20] to authenticate packet origins. There also exist several architecture proposals that aim to enable a DoS victim to stop unwanted traffic [30, 8, 22, 29, 24]. IPa+ is a more comprehensive architecture as it also provides secure routing and source authentication.

5. CONCLUSION

We present the design and a preliminary evaluation of IPa+, an accountable Internet architecture that is deployable on the IP network. IPa+ uses the rapidly deployed infrastructure DNSSEC to securely bind an address prefix to an authorized AS's public key, and uses the hash of an AS's public key as its self-certifying AS identifier in BGP. It uses this secure prefix-to-key binding to authenticate an AS's routing announcements and packets originated from its networks, preventing address prefix hijacking and source address spoofing attacks. Authenticated source addresses further enable simple DoS solutions that block attack traffic near its sources and prevent malicious flows from monopolizing a link's bandwidth to starve legitimate communications. We analyze the feasibility of IPa+, and compare it with AIP, an all-encompassing accountable Internet architecture that uses self-certifying addresses. The comparison shows that IPa+ provides nearly equivalent accountability to AIP, suggesting that IP can be made accountable without a major overhaul.

Acknowledgments

We thank Eric Osterweil for his insightful comments on an early draft of this paper. We also thank the anonymous reviewers for their feedback, and Garrett Wollman for the useful discussions. This work is supported in part by NSF Awards CNS-0627787 and CNS-0627166.

6. REFERENCES

[1] IEEE Standard 802.1X. `http://www.ieee802.org/1/pages/802.1x.html`, 2001.
[2] DNSSEC Trust Anchors from ARIN. `https://www.arin.net/about_us/dnssec/trust_anchors.html`, July 2009.
[3] Current State of Allocations and Assignments of Internet Number Resources. `ftp://ftp.arin.net/pub/stats/`, 2009.
[4] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable Internet Protocol (AIP). In *ACM SIGCOMM*, 2008.
[5] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, 2005.
[6] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, 2005.
[7] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, 2005.
[8] K. Argyraki and D. Cheriton. Active Internet Traffic Filtering: Real-Time Response to Denial-of-Service Attacks. In *USENIX*, 2005.
[9] F. Baker and P. Savola. Ingress Filtering for Multihomed Networks. RFC 3704, 2004.
[10] M. A. Brown. Pakistan Hijacks YouTube. `http://www.renesys.com/blog/2008/02/pakistan-hijacks-youtube-1.shtml`, 2007. renesys blog.
[11] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *SIGCOMM*, 1989.
[12] DNS Deployment Initiative. `http://www.dnssec-deployment.org/`, 2009.
[13] 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.
[14] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing. RFC 2827, May 2000.
[15] V. Fuller and T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. RFC 4632, 2006.
[16] Y. Hu, A. Perrig, and M. Sirbu. SPV: Secure Path Vector Routing for Securing BGP. In *ACM SIGCOMM*, 2004.
[17] 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.
[18] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (S-BGP). *IEEE JSAC*, 2000.
[19] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang. SAVE: Source Address Validity Enforcement. In *IEEE INFOCOM*, 2002.
[20] X. Liu, A. Li, X. Yang, and D. Wetherall. Passport: Secure and Adoptable Source Authentication. In *NSDI*, 2008.
[21] X. Liu and X. Yang. NetFence: Preventing Internet Denial of Service from Inside Out. `http://www.cs.duke.edu/nds/ddos/netfence-wip.pdf`, 2009.
[22] X. Liu, X. Yang, and Y. Lu. To Filter or to Authorize: Network-Layer DoS Defense Against Multimillion-node Botnets. In *ACM SIGCOMM*, 2008.
[23] E. Osterweil, M. Ryan, D. Massey, and L. Zhang. Quantifying the Operational Status of the DNSSEC Deployment. In *IMC*, 2008.
[24] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y.-C. Hu. Portcullis: Protecting Connection Setup from Denial-of-Capability Attacks. In *ACM SIGCOMM*, 2007.
[25] C. Sang-Hun and J. Markoff. Cyberattacks Jam Government and Commercial Web Sites in U.S. and South Korea. New York Times, July 2009. `http://www.nytimes.com/2009/07/09/technology/09cyber.html?_r=1&hp`.
[26] M. Shreedhar and G. Varghese. Efficient Fair Queueing Using Deficit Round Robin. In *ACM SIGCOMM*, 1995.
[27] T. Wan, E. Kranakis, and P. van Oorschot. Pretty Secure BGP (psBGP). In *NDSS*, 2005.
[28] R. White. Securing BGP Through Secure Origin BGP. *The Internet Protocol Journal*, 2003.
[29] A. Yaar, A. Perrig, and D. Song. SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks. In *IEEE Security Symposium*, 2004.
[30] X. Yang, D. Wetherall, and T. Anderson. TVA: A DoS-limiting Network Architecture. *IEEE/ACM ToN*, 16(6), 2008.