Evaluating the Performance Impact of PKI on BGP Security

Meiyuan Zhao*and Sean W. Smith
Department of Computer Science
Dartmouth College

David M. Nicol
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

February 2005

Abstract

The Border Gateway Protocol is central to making the Internet work. However, because it relies on routers from many organizations believing and passing along information they receive, it is vulnerable to many security attacks. Approaches to securing BGP typically rely on public key cryptography, in various encodings, to mitigate these risks; to work in practice, these approaches usually require public key infrastructure. This cryptography and the PKI may both potentially impact the performance of this security scheme; however, evaluating how these effects may scale to large networks is difficult to do analytically or empirically.

In this paper, we use the tools of simulation to evaluate the impact that signatures, verification, and certificate handling have on convergence time, message size, and storage, for the principal approaches to securing BGP.

1 Introduction

By distributing and maintaining routing information, the *Border Gateway Protocol (BGP)* [32, 39] plays a central role in making the Internet work. However, BGP relies on hearsay information. BGP speakers trust the messages they receive and they completely trust other BGP speakers to follow the protocol specification reliably. Consequently, BGP—and the Internet it routes—is vulnerable to many potential attacks by malicious players [26]. To mitigate these risks, many researchers have proposed security mechanisms to authenticate the routing information transferred between BGP speakers [1, 8, 13, 17, 35, 40, 41]. *S-BGP* is the dominant scheme here.

Because of the need to authenticate information passed among parties spanning a large set of domains, these security mechanisms typically rely on public key cryptography. Implicitly or explicitly, public key *infrastructure* thus also becomes a critical component—otherwise, how do the parties know what public keys to use and whether they are still valid?

Neither public key cryptography nor public key infrastructure come for free. However, when designing and analyzing these large information-distribution systems, it's easy to overlook these implementation details, and the performance impact they can have on the overall protocol. Furthermore, given the large, messy nature of Internet routing, it can be hard to evaluate this impact: analytic techniques may fail to capture the complexity, and empirical techniques may require a prohibitively large testbed.

In previous work [27], we used the tools of *parallel simulation* to evaluate the performance impact of basic signing and verification on route attestations—and proposed and evaluated an improved way of generating and encoding this information. In this paper, we extend this work:

- to consider two new aspects of performance: *message size* and *memory cost*;
- to consider the PKI impact of recent proposals for in-band *origin authentication*;
- to consider the performance impact of standard PKI *revocation* schemes; and
- to consider the potential improvement of using recent *aggregate signature* schemes in place of standard signatures in assertion chains.

We find that among the half dozen techniques studied there is no clear best solution. Compared to the technique that uses the least memory, the technique that supports

*contact author, zhaom@cs.dartmouth.edu

the fastest convergence time is three times faster but uses twice the memory. Signing cost is what matters for speed (and BGP convergence) but this comes at a price, memory and message size.

This Paper Section 2 reviews BGP and S-BGP. Section 3 reviews some alternate encoding and cryptographic approaches. Section 4 presents our evaluation methodology. Section 5 presents our experiments and results for path authentication. Section 6 presents our experiments and results for origin authentication. Section 7 reviews related work, and Section 8 concludes with some thoughts for future research.

2 BGP and S-BGP

The Border Gateway Protocol (BGP) [32, 39] is the routing protocol for maintaining connectivity between *autonomous systems (ASes)* in the Internet. Each AS is assigned a unique integer as its identifier, known as its *AS number*. An AS manages subnetworks expressed as IP *prefixes*—a range of IP addresses. A BGP *speaker*—a router executing BGP protocol—constructs and maintains *forwarding tables* that enable packet forwarding. A BGP speaker maintains connections with neighboring speakers, known as its *peers*, and sends an **Update** to announce a new preferred route to prefix p. The route is a `(prefix, AS path)` tuple. The *AS path* is a sequence of AS numbers that specifies a sequence of autonomous systems through which one can traverse the network; last AS in the sequence is the *originator* of this route. For instance, if the autonomous system AS_k owns IP prefix p, the autonomous system AS_0 might send out an **Update** $(p, \{AS_0, AS_1, \ldots AS_k\})$ to announce its preferred route to p. Each BGP speaker keeps received routes in its *routing table*; for each prefix, the speaker tags one route as its preferred one.

Typically, a speaker's routing table changes when it adds a new route, deletes a preferred route, or replaces a previously preferred route with a new one. BGP speakers incrementally send **Update** messages to announce such changes to their peers. When speakers establish (or reestablish) a *BGP session*, they share their own routing table with each other via a large number of **Update** messages announcing routes in their routing tables. If it results in new preferred routes, processing of an **Update** message may create a number of new **Updates**. If the speaker chooses to announce a new preferred route, it extends the existing AS path by perpending its AS number to it and sends it to all of its peers, except the one who sent the route earlier. When a speaker announces a route to prefix p, it implicitly *withdraws* the last route it announced to p. The recipient, understanding this implicit route withdrawal, decides whether it prefers the new route. A withdrawal can also be an explicit announcement, with no mention of an alternative preferred route. In this case, the recipient may examine the previously received routes to the same prefix and decide whether there is an alternative to announce to its peers. If no such route found at hand, it simply withdraws the route as well.

BGP rate-limits the sending of **Update** messages with parameter called the *Minimum Route Advertisement Interval (MRAI)*, which is basically the minimum amount of time that must elapse between successive batches of **Updates** sent to a neighbor. BGP speakers have output buffers to keep waiting **Update** messages, and send them in batches when reaching the MRAI. A speaker may have a different MRAI for each of its peers or may have one MRAI that controls all peers. In practice, throughout the Internet, the default value the MRAI is 30 seconds.

Any change of network reachability will be reflected in the routing table of some BGP speaker. BGP will then propagate this change via **Update** messages through the entire network, like a wave. The *convergence time* measures the length of time for such wave of announcements to die out completely—in other words, for the network to return to a stable state. During the transient period of convergence, the continual changing of preferred routes degrades the effectiveness of packet forwarding. Longer convergence times thus reflect increased network instability and may cause severe network performance problems. Studies of BGP have considered convergence [10, 20, 34] and possible optimizations to control and accelerate it [11, 19, 21, 23, 30, 38].

Because BGP is central to Internet functionality and is vulnerable to malicious actors, we need to secure the information that BGP distributes. We consider each component:

- *Origin authentication* considers whether the originating AS really controls a claimed IP address range.
- *Path authentication* considers whether a claimed path to reach some IP prefix is in fact valid.

The dominant security solution, *Secure BGP (S-BGP)* [17] focuses on the **Update** messages. The first step of S-BGP is to set up public key infrastructures to help establish the authenticity of the involved parties. S-BGP uses X.509 [12] public key certificates and puts BGP-related information into certificate extensions. Speakers digitally sign the **Update** messages they announce to peers; with these X.509 certificates, recipients can verify the signatures to authenticate the received routes.

More specifically, each speaker uses *address attestations (AAs)* for origin authentication, and *route attestations (RAs)* for path authentication.

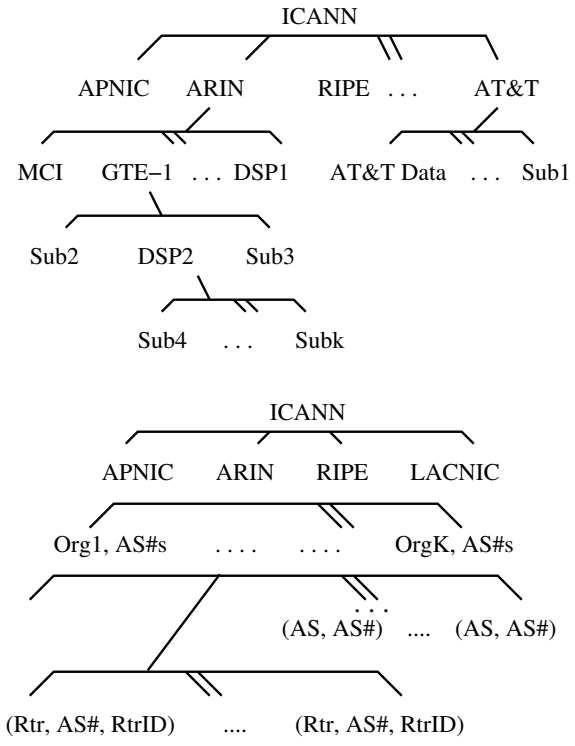

Figure 1: Sketch of the S-BGP PKIs.

2.1 S-BGP PKIs

To enable validation of attestations, S-BGP proposes two X.509 public key infrastructures. The first PKI contains certificates to authenticate the *owners of portions of the IP address space*. The second PKI is to authenticate BGP speakers, ASes, and the owners of ASes. Figure 1 illustrates the structures of these PKIs. Both PKIs are hierarchies rooted at ICANN [15]. ICANN issues itself self-signed certificates and further issues certificates to the the first tier of organizations, typically *Regional Internet Registries (RIRs)* such as ARIN, RIPE, APNIC, and LACNIC.

For the address allocation PKI, ICANN issues itself a certificate claiming the ownership of entire IP address space on the Internet. Consequently, it issues certificates to RIRs as it assigns IP address blocks to them. The certificate contains an extension that specifies the set of address blocks ICANN is allocating to that RIR. Each RIR further assigns portions of its address blocks and issues corresponding certificates to the third tier organizations of the hierarchy. The process continues until it reaches end subscribers. A typical certification path for an address block is similar to the following:

"ICANN→Registry→ISP/DSP. . . →Subscribers".

The second PKI contains certificates for AS number assignments, as well as identity certificates of organizations, ASes, and BGP speakers. The AS number authentication is similar to address allocation authentication. At the top, ICANN assigns AS numbers to RIRs. Then, each RIR assigns some of its AS numbers and issues certificates to the third tier organizations (also called AS owners). These AS owners, in turn, issue certificates for authenticated ASes. AS owners also issue certificates for BGP speaker; each such certificate binds the router name to an AS number and router ID, testifying that the speaker belongs to certain AS. Typical certification paths in AS number and BGP speaker identification PKI are as follows:

"ICANN→Registry→AS owners→AS numbers"
"ICANN→Registry→ISP/DSP. . . →BGP speakers".

2.2 S-BGP Attestations

As noted earlier, S-BGP uses two forms of attestations.

For origin authentication, an address attestation (AA) establishes that an AS (the subject in the AA) is authorized by an organization Org_x (the signer of the AA) to announce certain IP blocks of address space [17]. The origin AS sends the AA together with a certificate that authorizes that Org_x in fact owns that IP address block. Hence, the receiver of the **Update** message is able to validate the certificate and verify the signature in this AA.

For path authentication, a *route attestation (RA)* is signed by a BGP speaker to authenticate the existence and position of an AS number in an AS path [17]. Figure 2 demonstrates the structure of RAs. Such attestation is nested: each BGP speaker signs the AS path in sequence, as it joins. That is, first the origin BGP speaker signs the AS number of the origin autonomous system and the intended receiver (in the form of AS number). The next signer is the receiver of this RA; it computes and signs

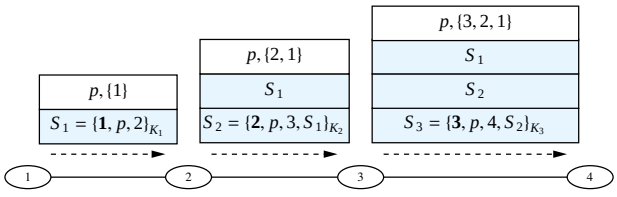

Figure 2: This figure sketches the process of sending route announcements and their route attestations. We have four ASes numbered as `1`, `2`, `3`, and `4`. `AS 1` initiates the process by sending announcement `(p, {1})` stating that it owns prefix p and it is reachable. It generates the corresponding route attestation by signing `{1, p, 2}` using its private key K_1. It puts its AS number first, then the prefix, then the intended recipient. The other ASes continue this process, except that they glue new information to the previous attestation sign the resulting blob. The figure shows the AS path components in bold.

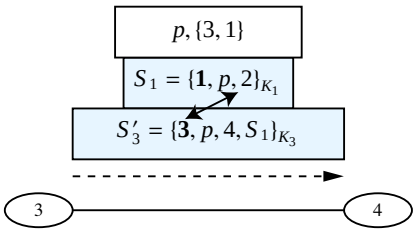

Figure 3: This figure sketches how S-BGP would stop an attempt by `AS 3` to forge a route announcement. `AS 1` had told `AS 2` it would accept messages to *p*, and `AS 2` told that to `AS 3`. However, `AS 3` is trying to strip away `2` and fool `AS 4` into believing a fraudulent 2-hop route. However, since `AS 1` included the name of `AS 2` in its signed statement about that link, `AS 4` will detect the forgery.

the concatenation of the previous RA, the newly appended AS number, and intended receiver. The process goes on until the entire AS path is signed.

The inclusion of the intended recipient and the prefix in each signature is necessary to prevent against "cut-and-paste" attacks. To continue the earlier example, consider Figure 3. `AS 3` is not able use the attestations it has received to forge an attestation for route `(p, {1,3})` that `AS 4` will accept. To do so, `AS 3` would need a signed statement from `AS 1` offering to route information to *p* directly from `AS 3`. However, the signed link that `AS 3` has from `AS 1` explicitly specifies that `AS 1` links to `AS 2`, not `AS 3`. To facilitate validation, BGP speakers send the new RA together with all the nested RAs associated with it. This way, the receiver can authenticate the entire AS path. However, receivers need certificates for BGP speakers to validate these signatures.

2.3 Performance Issues of Path Authentication

Several factors affect the performance of path authentication in S-BGP, given the structural properties of RAs.

First, BGP speakers consume extra CPU cycles when signing and verifying RAs and when handling and validating certificates. Each **Update** message involves one signing operation by each signer and *k* verification operations by each verifier (where *k* is the number of RAs for this AS path). Moreover, for each signature verified, the verifier needs to validate the certificate of the alleged signer. Second, RAs and certificates increases message size. Each message with an AS path of length *k* carries *k* nested RAs. Finally, to decrease the number of signing/verification operations, one could cache the signed or/and verified routes in memory. Therefore, memory cost becomes another issue.

Researchers have introduced a number of optimizations for S-BGP [16], mainly focusing on caching signed and verified routes and applying DSA pre-computation. These optimizations reduce the computational cost related to cryptographic operations in the cost of extra memory cost and computation complexity.

3 Alternate Signature Approaches

Besides caching, other studies suggest different cryptographic schemes that may potentially reduce the overhead of S-BGP route announcement authentication. We discuss three: signature amortization, sequential aggregate signatures, and origin authentication.

3.1 Signature Amortization

In our previous analysis [27], we proposed *Signature Amortization* (S-A).

Looking at the details of the path authentication process, we observed two important facts. First, BGP speakers verify RAs more often than creating RAs themselves. Hence, making verification faster could potentially decrease the overall computational latency. Second, when the BGP speaker sends identical routes to its neighbors, it has to create distinct RAs; moreover, BGP speakers keep outgoing **Update** messages in buffers and, using MRAI timers, send them in bulk. This bulk send creates the potential for getting more "bang" from each private key operation.

Our S-A scheme exploits these two facts. To speed up the verification processing, we use RSA, since RSA verification is significantly faster than DSA (used by S-BGP). Then, we amortize the cost of signing operation in two steps.

In step one, when a BGP speaker sends the same route announcement to multiple recipients, we collapse it to literally the same announcement—using a bit vector (or a more space-efficient equivalent) to express which of the speaker's peers are the recipients. Thus, the speaker only needs to generate one signature, instead of one for each recipient; the verifier of this RA uses the bit vector to check the intended receiver. To do this, the speaker needs to preestablish an ordered list of its neighbors, and distribute this to potential verifiers; however, we can put this information in the speaker's X.509 certificate, since the verifier needs to obtain that anyway to verify the signature itself.

In step two, when its MRAI timer fires and a BGP speaker sends the messages accumulated in its out buffers, we have it collect all "unsigned" messages, build a Merkle hash tree [24, 25] on them, and signs the root of the tree—

thus generating one signature for all unsigned messages, instead of one for each. A leaf of the tree is the hash of the pair of a route and its recipients. The resulting RA consists of the RSA signature on the root, the the hash path from the root to that leaf, the route, and the recipient bit vector. A verifier of the RA can use these hash values and information in the route announcement to construct the root of the tree correctly. There are trade-offs, however. The verifier needs to perform a few extra hashing operations when verifying a RA, and the message size grows (due to the hash path).

With our S-A approach, we speed up the security operations of S-BGP at the cost of more memory and longer **Update** messages.

3.2 Sequential Aggregate Signatures

Recently, *aggregate signature* schemes have emerged that save signature space when multiple parties need to sign messages [2, 3]. The *sequential aggregate signature (SAS)* scheme by Lysyanskaya et al. [22] combines n signatures from n different signers on n different messages into one signature of unit length. In SAS, each signer, in an ordered sequence, incrementally signs its new message and incorporates it into the aggregate signature σ. A party with knowledge of the n messages, the public keys of the n ordered signers, and the final σ is able to verify that each signer s_i has correctly signed his message M_i and σ is a valid sequential aggregate signature. The major advantage is that the signature of n messages is the same as the length of an ordinary signature. Furthermore, an SAS scheme can be built from RSA, with small modifications, easing implementation.

Applying SAS scheme to path authentication of S-BGP, we generate σ along the AS path similar to nested RA signatures. Since one aggregate signature is enough to authenticate entire AS path, this scheme shortens message size.

3.3 Origin Authentication

Aiello et al. [1] consider the semantics, design, and costs of origin authentication in BGP, and propose an *OA* scheme.

The authors formalize semantics for IP address delegation, which is similar to the address allocation PKI by S-BGP. The proofs of the IP address ownership establish a tree-like hierarchy rooted at IANA [14]. The next tier are the organizations that receives `/8` IPv4 address blocks directly from IANA. These organizations further delegate sub-block addresses; delegations continue until we reach autonomous systems.

In the Aiello OA scheme, the BGP speakers send ordinary BGP **Update** messages together with *origin authentication tags (OATs)*. Each OAT contains a delegation path, a set of *delegation attestations* (one for each edge in the path) and an *ASN ownership proof*. The structure of a delegation attestation is similar to an S-BGP address allocation certificate. The signer authorizes that the subject is delegated some address blocks as recorded in an extension. The ASN ownership proof is a certificate issued by ICANN; it attests that some AS numbers are granted to a particular organization.

The OA scheme considered four possible constructions on delegation attestation. A *Simple Delegation Attestation* contains a signature by an organization on a tuple `(p, org)`, where `p` is the prefix delegated to `org`. An *Authentication Delegation List* combines all `(p, org)` tuples by the same organization into single list and generates one signature. A compromise of these two approaches, an *AS Authentication Delegation List* breaks up the long list into several sublists (each containing the delegation tuples specifying the address delegations made to the same organization and autonomous system) and signing each. An *Authentication Delegation Tree* constructs a Merkle hash tree on an organization's delegation list, and signs the root of the tree. We denote these variations by the terms *OA-Simple*, *OA-List*, *OA-AS-List*, and *OA-Tree*, respectively.

4 Evaluation Methodology

As Section 1 notes, this paper reports research examining the performance impact of public key cryptography and public key infrastructure on BGP security. Section 4.1 describes the metrics we use. Section 4.2 describes the various BGP security approaches on which we take these measurements. Section 4.3 discusses the tools we use to carry out these experiments.

4.1 Performance Metrics

We use a set of metrics to evaluate performance in terms of time and space.

For time, we measure the number of cryptographic operations involved, the resulting CPU cycles, and the BGP convergence time: the time it takes the system to reachieve a stable state after a perturbation, such as a new route announcement, a route withdrawal, or a router reboot. For each security scheme, we compare its convergence time with convergence time that original BGP achieves for the same perturbation. (Given the distributed nature of BGP, convergence time is very difficult to be predicted using analytical techniques.)

For space, we measure both the message size and the

storage cost in memory. Similar to other studies, our experiments relax the current BGP *maximum transfer unit (MTU)* (4096 bytes) limitation, to be able to understand the efficacy of any possible optimization.

4.2 Experimental Approaches

Our previous work evaluated the time impact of S-BGP and S-A on path authentication. We now measure the space impact as well, and both space and time impacts of SAS on path authentication. We measure the time impact of CRL and OCSP revocation schemes on fully optimized S-BGP.

We also examine the origin authentication scheme of Aiello et al. We measure time and space impacts of all four variations, as well as the time impact of CRL and OCSP revocation on the OA-AS-List variation (since it's the closest to S-BGP origin authentication).

4.3 Simulation

We use discrete-event simulation to understand the performance of BGP origin and path authentication schemes in a large-scale environment. As with our earlier work, our experiments uses SSFNet [5, 28], a discrete-event simulator that provides a comprehensive model of basic BGP operations [31]. Our earlier work added hooks for variants of processing models of BGP security schemes [27].

Throughout this study, we evaluate security schemes in the same network topology and same BGP activity settings. We use a 110-AS topology, with one operating BGP speaker per AS. For modeling simplicity, each BGP speaker announces two prefixes. In our model, each AS also possesses *virtual BGP speakers* that don't actually run a simulated BGP protocol. We use the number of such BGP speakers to represent the size of an AS; its size affects the time it takes for one **Update** message to be propagated through an AS.

We use the public data provided by RouteViews project [33] to generate a graph of AS connectivity of the Internet, then reduce the size to 110 ASes using a collapsing procedure. This reduced graph still preserves certain macroscopic properties [6] seen on the entire Internet. Moreover, we incorporate our estimation of route filtering policies into the topology using a method, similar to the one proposed in [7].

During normal BGP activities, we let one BGP speaker crash and reboot. We evaluate the performance of the entire system during router rebooting process. The workload on BGP speakers could be much higher than normal BGP activities, since the rebooting BGP speaker receives routing table dumps in a short period of time from each its peers, via a large amount of route announcements. To maximize the effects, we let the rebooting BGP speaker to be the one with the most peers.

	Convergence	Message Size	Memory
S-BGP	long	moderate	best
S-A	shortest	worst	worst
SAS	longest	best	best

Table 2: Performance rankings for the path authentication schemes we examined

Besides the common settings, we also have specific parameters for each of the security schemes. Table 1 summarizes the benchmarks and measurement numbers we use in our simulation. The running time benchmarks of cryptographic operations are from OpenSSL [29] library. For those algorithms not directly implemented by the library (such as DSA pre-computation, SAS aggregate signing and SAS aggregate verification), we decompose the involved operations and sum up the benchmarks of each step as an estimation. In addition, the numbers are normalized to a 200MHz CPU, which is a common CPU speed of BGP routers. We use a real system to measure and estimate latencies of processing plain **Update** messages, of sending a OCSP request and receiving a response, and of fetching CRLs. To take into account other factors that could potentially affect the numbers, the simulation decides these values by uniform distribution within certain ranges. S-BGP studies [16, 18] give the numbers for sizes of S-BGP certificate and attestations.

5 Path Authentication Performance Analysis

We compare performance impact of S-BGP, S-A, and SAS. We examine the performance on signatures and PKIs respectively. This section gives detailed results and analysis.

5.1 Signatures and Verifications

Before examining details, we enumerate our major findings on convergence time, message size, and memory cost in Table 2. S-BGP performs badly on convergence time, but is fairly efficient on memory cost. S-A outperforms the other two on convergence time, but is significantly more costly than the other two schemes on message size and memory cost. SAS generates the shortest **Update** messages, but results in the longest convergence time.

We also studied the efficacy of strategies for caching validated (or generated) signatures. In simulation experi-

	SHA-1 hash	MD5 hash	Attestation	S-BGP X.509 Certificate	Identifier
Length (bytes)	20	16	110	600	4

	RSA	DSA	DSA (p-c)	SAS
Verify Time (*ms*)	2.5	31.0	31.0	2.5
Sign Time (*ms*)	50.0	25.5	0.015	50.0
Signature Length (bytes)	128	40	40	128

	OCSP request	CRL fetching
Operation Latency (second)	0.5–1.0	0.5–1.0

Table 1: Constants and benchmarks used for simulation. RSA, DSA, and SAS algorithms are based on 1024-bit keys.

ments, we explored S-BGP with several variations of DSA optimizations. For the presentation of experiment results, we use *cDSA* to denote S-BGP with caching, *pDSA* to denote S-BGP using DSA pre-computation, and *cpDSA* for S-BGP with both optimizations. In our model, these caching strategies store both validated signatures and generated signatures; we use $10\mu s$ (with a uniformly distributed delta of $5\mu s$) to model the lookup time. The S-A scheme will not speed up by caching hash trees with signatures, because the trees, and hence the signatures, are constantly changing even for the same route announcement (since the trees depend on the context of what else is being signed at that time). Therefore, we only examined S-A scheme without caching, when studying processing latency and convergence time. However, we model a special variation for S-A caching merely to understand potential memory cost it might result. Finally, all the experiment results are average numbers from 20 runs of the simulation. The standard deviation is less than 5%.

Time We examine the convergence time by looking at the counts of cryptographic operations. Figure 4 through Figure 7 summarize the results. All the schemes without caching optimization generate relatively the same number of signature verifications, proportional to the total number of AS numbers encountered in AS paths in route announcements. Similarly, caching optimization by each of the schemes achieves relatively the same number of saving percentage.

The story of signing operations remains the same for S-BGP and SAS schemes. The S-A scheme can dramatically save as many as 98.3% of signing operations. Our experiments show that the average hash tree size by S-A is about 60.1, indicating that S-A is able to amortize the cost of 60 signing operations into only one signing and a few hashing operations.

The CPU cycles and convergence time reflect this difference in the number of cryptographic operations. We sum up the total CPU time on all BGP speakers, and also track the portion consumed by cryptographic operations, including signing, verification, and hashing ("crypto," in Figure 6). SAS requires 1723.2 seconds extra time for aggregate signing and aggregate verification, which is much shorter than 4002.2 seconds by S-BGP. This difference results mainly because aggregate verifications are much faster than DSA verifications. Caching optimization to S-BGP and SAS scheme can significantly reduce total CPU time. Although S-BGP (pDSA) uses much faster signing operations, the net speed-up is limited, because the number of verification operations dominates the number of signing operations. Compared with S-BGP and SAS, S-A improves both aspects—fewer signing operations and faster verifications. Our experiments confirm it is the most efficient on CPU cycles.

Next, we look at convergence time. Among the three major schemes, SAS is the worst. Compared with plain BGP, it converges three times slower. S-BGP comes next, with a slowdown of about 2.3 times. Even with optimizations, S-BGP still takes 46.05% longer to converge. (Our previous work [27] showed better S-BGP numbers, but that turned out to be due to a bug in our simulation code.)

Such slowdowns lead to routing fluctuations that create all sorts of network problems, such as increased packet loss rates, increased network latencies, increased network congestion, and even disconnections. In our experiments,

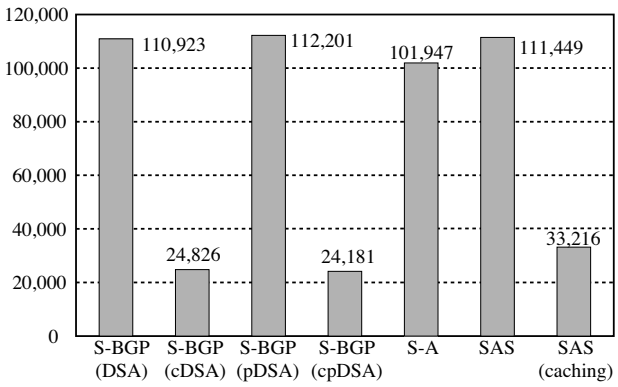

Figure 4: Verification operations in path authentication

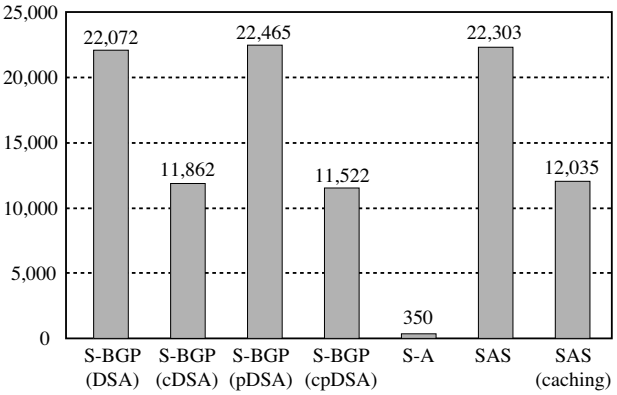

Figure 5: Signing operations in path authentication

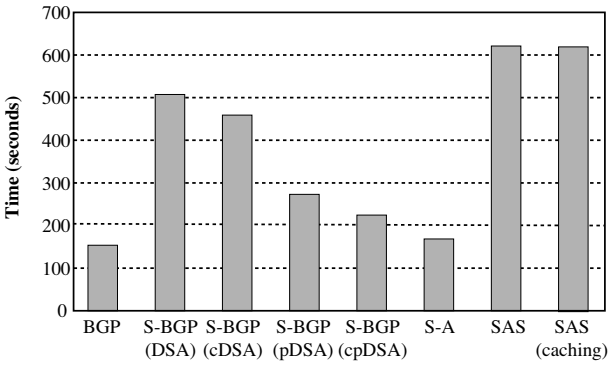

Figure 7: Convergence time in path authentication

router reboots by BGP even without any security protection already cost the network 153.7 seconds to converge. Extending the period to another several minutes is not a good option.

Fortunately, our S-A scheme increases the convergence only by a few seconds, with no burden on caching large amount of data in memory.

Our experiments revealed that, counter-intuitively, convergence time is not proportional to the CPU time spent by BGP speakers. In fact, the data suggests that the latencies in the message sending process (therefore, signing overhead) could be the dominant factor. For instance, if we consider only the CPU time consumed by signing operations, SAS costs the most, about 92% of the total CPU time, which could explain why SAS is the slowest on convergence. One might reach a similar conclusion from the inconsistency between S-BGP (cDSA) and S-BGP (pDSA). Although S-BGP (pDSA) requires more CPU cycles, almost all of these CPU cycles are spent for signature verifications. As the result, it converges faster.

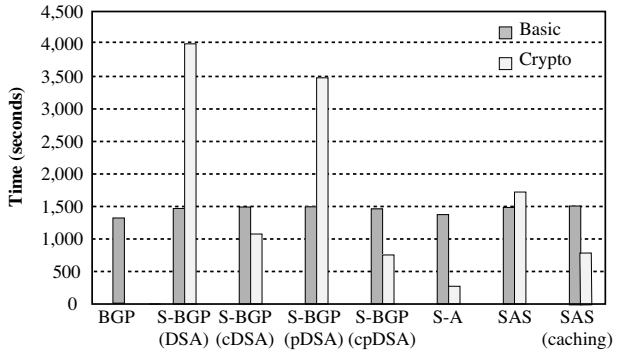

Figure 6: Total CPU time in path authentication

Memory Figure 8 shows the average memory cost and maximum memory cost for individual BGP speakers. We start with a baseline of 9KB memory at each speaker, for plain BGP. On average, S-BGP increases this requirement to 112.25KB, SAS to 121.95KB, and S-A to 314.38KB. We assume that BGP speakers record all cached routes in memory (e.g., RAM). In the simulation, we count the bytes of the IP prefix, AS path, and related cryptographic data structures (signatures, hash values, and bit vectors).

As mentioned earlier, frequent changes of hash trees prevent S-A from saving processing latency by caching signatures. To explore the memory impact of caching, we tried letting S-A cache more stable data, the leaf information: **Update** messages, signatures, and associated bit vectors (assuming neighboring relationship between ASes stays unchanged during simulation). For this experiment, we dispensed with hash trees, but the resulting convergence time of a variant that used hash trees and this caching would not be worse than the numbers shown in Figure 7 and 8.

One of the leading factors that affect this memory cost

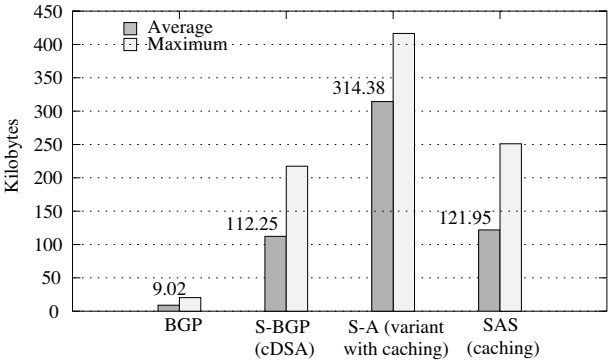

Figure 8: Comparison of memory costs for caching. The S-A scheme in this comparison is a variant that does not use hash trees, and caches leaf information instead of signatures.

	BGP	S-BGP	S-A	SAS
Average message size (bytes)	36.09	318.61	1107.08	184.29
Increase		8.83	21.57	5.11
Maximum message size (bytes)	42.6	527.7	1915.4	191.2
Increase		13.77	47.75	4.40

Table 3: Message size. The increase numbers are based on the message size by original BGP.

is signature length. Here, S-BGP outperforms S-A because a DSA signature is much shorter than a RSA signature (e.g., 40 bytes vs. 128 bytes). Secondly, SAS is able to save memory by caching only one signature for an AS path of any length. Even with RSA signatures, SAS is as efficient as S-BGP.

Although not shown in Figure 8, edge routers consume the most memory for caching routes, statistically. We posit two reasons. First, as a pure customer in the network, an edge router may receive more route announcements than the ones in the core of the network. Second, and most importantly, the AS paths recorded by edge routers are significantly longer, so these routers will cache more signatures.

In ongoing work, we are exploring using cryptographic hashing to further reduce cache size.

Update Message Size SAS produces one signature for an AS path; it wins the competition on message size. S-BGP is next, again, because of shorter signature length. Our experiment results, shown in Table 3, confirm that S-A generates the longest messages. For both S-BGP and S-A, number of signatures in messages grows as the length of path increases.

For SAS, since each **Update** message contains only one aggregate signature for the entire AS path, the maximum message size is close to the average size. On the other hand, the longest **Update** message for the S-BGP and S-A schemes is about two times as long as average messages.

Our experiments measured shorter message sizes than the number measured in the Internet, because we only considered the fields (AS path, signatures, hashes, and bit vectors) that would vary between the schemes. Since the ignored portions are the same for each of the schemes, the simulation still results in a fair comparison of the message size.

5.2 Certificate Revocation

Bringing the PKI one step closer to reality requires considering the costs of checking the validity of a signer's certificate, when verifying a signature. Recall that BGP speakers use their private keys to sign and create RAs on route announcements. We use simulation to model the case that BGP speakers validate BGP speakers' public keys in certificates before using them to verify RAs.

In our revocation simulation, we assume that the 110 ASes belong to different organizations (also called PKI *domains*), with each organization having its own CA issuing certificates for that organization's BGP speakers. Each PKI domain has a repository of certificates, offered by an LDAP server. When we model revocation by OCSP, we assume an organization has an online OCSP responder; when we model CRLs, we assume the organization's LDAP server also offers CRLs.

We then examine the convergence time for S-BGP with all optimizations, using OCSP or CRLs for certificate validation. The OCSP approach provides fresh information of certificate status, at the cost of network and processing latencies. The CRL approach is less aggressive: the verifier downloads CRLs periodically, checks certificate status with these local copies, and (when the local copies expire) get fresh CRLs from the appropriate repositories via the LDAP protocol.

For simplicity, we assume that BGP speakers can validate OCSP responses and fetched CRLs by verifying signatures on them. In other words, we do not model the process of discovering trust path for them. The rest of this section discusses and compares the performance impact that checking certificate status has on S-BGP.

OCSP The model we use to study OCSP is close to typical PKI practice in the real world. In a practical PKI, one or more OCSP responders connect to a certificate database operated by local CAs to serve the status information of the certificates issued by local CAs. Optionally, the responders can set up SSL connections to enhance privacy for the client.

The OCSP response is a signed data structure that contains the real-time status of a requested certificate. OCSP introduces latencies, from setting up an SSL connection, from network delays, from real-time signing, and from signature verification. According to measurements we made with real-world OCSP implementations, the latency of one round is about 0.5–1.0 seconds, the majority of which is from network latency.

If a client has multiple certificates to validate, it can send OCSP requests in sequence or in parallel. A proxy, such as a *Certificate Arbitrator Module* (CAM) [37], can

Protocol	# Ann.	# Vrf.	# Sign	# OCSP Rqst.	Basic CPU (s)	Crypto CPU (s)	Convergence (s)
BGP	19571.8	–	–	–	1310.6	–	153.7
S-BGP (cpDSA)	21898.9	24180.6	11521.9	–	1464.1	755.4	224.4
Sequential OCSP	22542.9	113859.9	11663.2	89912.5	1501.7	70990.2	2720.4
Parallel OCSP	21707.8	110429.3	11290.5	87004.0	1448.5	3971.0	344.3

Table 4: Performance of validating certificates using OCSP for S-BGP path authentication

contact multiple OCSP responders throughout the network and send requests in parallel for the client. In our simulation, we model both sequential and parallel cases.

Table 4 shows that checking certificate status using OCSP for S-BGP is intolerably expensive. Sending sequential OCSP requests is an especially bad idea. We put performance numbers of BGP and S-BGP (cpDSA) in the table for comparison, and show both the basic CPU time, the processing latencies related to cryptographic operations, the latencies by OCSP requests and responses, and network latency in between. Even the resulting convergence time for parallel OCSP requests is 344.3 seconds.

CRLs For CRLs, we assume that each BGP speaker has a local cache of CRLs. Since signature verification requires an up-to-date copy of the CRL from the relevant CA, the BGP speaker pays the price of fetching and validating fresh ones before verifying RAs, if some CRLs are missing or expired.

To evaluate the cost of fetching CRLs, we let BGP speakers have a certain fraction of the CRLs in their local cache be expired, and then measure the resulting convergence time. The experiments assume that it costs 0.5–1.0 seconds on average for BGP speakers to fetch a CRL. We also assume that CRLs are valid for 12 hours.

Figure 9 shows the measurement data from simulation. It is clear that more expired CRLs cause the convergence times to increase linearly. These times range from 224.4 seconds to 287.7 seconds. Hence, even with all CRLs expired, validating certificates against CRLs is still a more efficient approach than OCSP, which costs 344.3 seconds to converge with the fast option, parallel OCSP requests.

6 Origin Authentication

Our approach to studying origin authentication is similar to the approach we took for path authentication. We first look at the performance impact of signatures and verifications, then examine the certificate validation cost on top of that. We add one model in simulation for experiments—the approximated address delegation graph. As mentioned earlier, the semantics of IP address delegation start from IANA. Aiello et al. [1] expressed IP addresses of the Internet using such semantics. Using publicly available Internet measurements, these researchers generated an approximated address delegation graph, a tree rooted at IANA. The structure is very similar to the address allocation PKI by S-BGP (not surprising, since it essentially solves the same problem).

For each prefix in route announcements, the **Update** message should carry an address delegation path for authentication. The scheme of Aiello et al. [1] uses in-band address delegation attestations carried in **Update** messages, because these attestations are much smaller in size than the S-BGP address allocation certificates. We use simulation to re-visit this issue.

We model address delegation using this approximated complete graph of the Internet and size it down so that it is suitable for our 110-AS simulated network. In practice, ASes could announce many prefixes, each of which could have its own address delegation path in the graph. Our simulation model is much simpler; each AS only announces two prefixes. We add randomness in the model to capture the diversity of the real world. First, we put the address delegation graph into the configuration of simulation, so that BGP speakers can recognize all delegation paths for each origin AS. Next, we let BGP speakers randomly choose a path for the prefix based on the origin AS. We limit the path length to seven (since address delegation paths are reported to be no longer than 4-5, in practice).

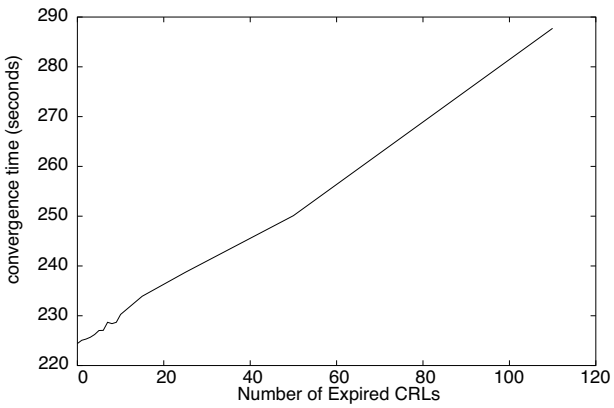

Figure 9: Convergence times by S-BGP using CRLs to validate certificates.

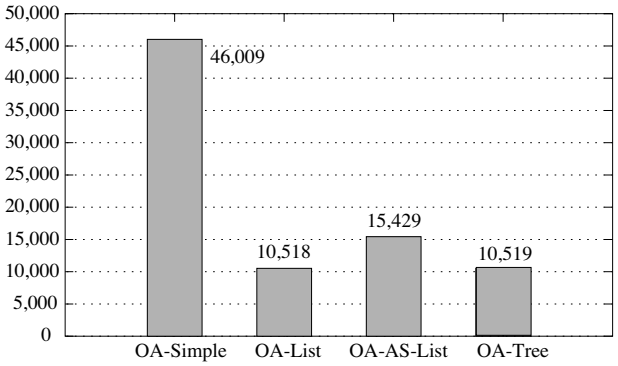

Figure 10: Number of verification operations by OA address delegation attestation constructions.

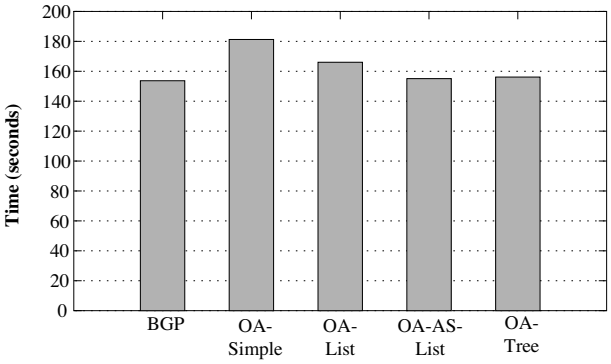

Figure 12: Convergence time by OA address delegation attestation constructions.

This randomly chosen path determines what address delegation attestations are involved.

6.1 Signatures and Verification

Time Figure 10 through Figure 12 show the processing latency. We assume that organizations prepare the delegation attestations offline; the simulation only counts signature verifications and hashing latencies accordingly. The OA-List and OA-Tree approaches greatly reduce the number of signature verifications required. Compared with path authentication schemes, the increase of convergence time by all delegation attestation constructions are manageable. This result, again, implies the verification overheads are a minor factor to convergence time compared to signing operations.

The resulting convergence time of OA confirms the conclusion made by Aiello et al. [1]—the efficiencies afforded by OA designs make in-band delegation attestation verification possible. However, as Aiello et al. also mention, in-band delivery of delegation attestation is susceptible to replay attacks, unless we introduce short-lived tokens or make delegation attestations short-lived. Thus, a trade-off exists between the period of vulnerability and the overhead of administration and computation.

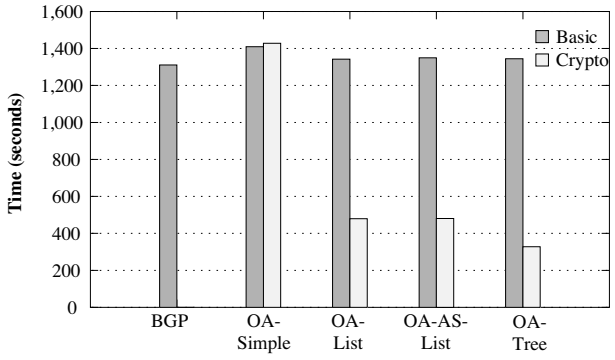

Figure 11: Total CPU time by OA address delegation attestation constructions.

Memory We let BGP speakers cache verified attestations and associated prefixes; we then measure the average memory cost and message size. Table 5 shows that the OA-List scheme is more costly than other schemes, mainly because the list construction produces extremely long delegation attestations. In the approximated address delegation graph, the average number of delegations made by organizations is about 56.96. Moreover, about 16 organizations make 80% of the address delegations. Obviously, this graph has high connectivity and the delegations are concentrated on very small portion of organizations. These features are the reason why the AS-List approach can produce long lists of prefixes in address delegation attestations. According to Figure 10, the AS-Tree approach handles the least number of signatures; however, its memory cost and message size are worse than OA-AS-List, mainly because the AS-Tree approach involves hash values, which are much longer than organization identifiers.

6.2 Certificate Revocation

The above analysis shows that the OA-AS-List attestation construction is fairly efficient. It is the most efficient one on memory cost and message size, and it does not put significant pressure on BGP processing and convergence. In fact, the OA-AS-List construction—the delegation list grouped by different delegatees—is very similar to the design of address allocation certificates of S-BGP. Thus, we next consider the case that BGP speakers send S-BGP ad-

Attestation Constructions	OA-Simple	OA-List	OA-AS-List	OA-Tree
Storage for Attests. (KB)	42.80	666.27	13.23	30.22
Message Size (Bytes)	496.97	36293.37	575.35	1029.24

Table 5: Average memory cost and message size by OA address delegation attestation constructions.

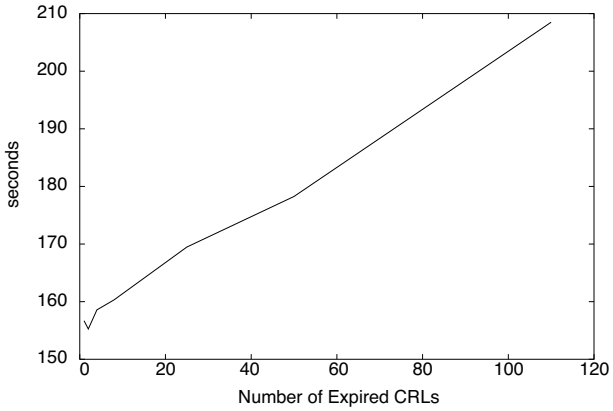

Figure 13: Convergence times by origin authentication using CRLs to check certificate status.

dress allocation certificates, instead of delegation attestations in **Update** messages. In other words, the sender encloses the complete certification chain for the verification of address attestation (AA). We assume that each speaker sets ICANN and the CA in their local PKI domain as its trust anchors.

Again, bringing this PKI one step closer to reality requires considering the costs of checking the validity of the certificates. We consider each approach in turn.

OCSP As before, we first consider OCSP, both in sequence and in parallel. Table 6 shows the experiment results on processing latency. The most important conclusion we can draw is that, as for path authentication, OCSP processing for origin authentication can greatly slow down the BGP convergence. For either part of BGP route authentication, using OCSP to validate real-time certificate status does not appear to be feasible in practice.

CRLs Again, we carried out experiments assuming different sets of CRLs expire at the routers, and examined performance. Figure 13 shows the results. The curve is similar to the convergence time by path authentication with CRL fetching. The convergence time is relatively unaffected if each of the BGP speakers needs to fetch fewer than eight CRLs during rebooting.

6.3 Certificate Distribution

In addition to processing latency and convergence time, the experiments also measure message size. Carrying certificates in **Update** messages would require 2KB on average. The maximum message size about 4KB. Given the BGP message MTU, carrying these certificates does not appear to be feasible in practice. On the other hand, if BGP speakers record all certificates locally, our simulation shows that certificates consume about 6KB storage on each BGP speakers, on average. The relatively small scale of the simulated network prevents us from directly inferring potential storage issues in the real world. IP address allocations, AS number assignments, and router assignments on the full Internet produce much more certificates. The CIDR BGP report from AS1221 (Telstra) [4] shows that there are 181,031 active BGP entries in a routing table. To validate ownerships of these prefixes, we need roughly the same number of address allocation certificates. Besides, this report also concludes that there are about 18,233 unique ASes and 50,000 organizations. Considering both PKIs by S-BGP, each BGP speaker needs about 190MB in total to store all certificates.

7 Related Work

The performance studies in [16, 18] offer detailed discussions on deploying S-BGP in the real world. The authors collected a variety of data sources to analyze S-BGP's performance impacts on BGP processing, transmission bandwidth, and routing table size. These studies concluded that the memory requirements of holding route information and related cryptographic data are a major obstacle to deployment of S-BGP. Unlike our work, all of the discussions are based on static measurement of BGP.

The origin authentication study by Aiello et al. [1] designed a simulator, *OASim*, to model the operations of a single BGP speaker. This simulator accepts timed BGP **Update** streams and computes the costs associated with the validation and storage of the related origin authentication proofs. The simulation results show that in-band distribution of origin authentication proofs is possible. Our simulation is more powerful than OASim in that we model and simulate a network and study the convergence time.

Our previous study [27] used a packet-level detailed

Protocol	# Ann.	# Vrf.	# Attest.	# OCSP Rqst.	Basic CPU (s)	Crypto CPU (s)	Convergence (s)
BGP	19571.8	–	–	–	1310.6	–	153.7
OA-AS-List	20131.2	15429.1	10364.1	–	1349.7	480.4	155.1
Sequential OCSP	22800.5	73586.7	5071.0	68515.65	1522.1	53665.7	2420.9
Parallel OCSP	22408.6	72635.2	5071.2	67564.00	1494.8	19060.2	938.7

Table 6: Convergence impact of OCSP on in-band address attestation.

simulation model of BGP to understand the processing overhead by S-BGP. We discovered that, due to public key cryptography, S-BGP is expensive on operational latency and thus greatly increases convergence time. We further proposed a more efficient scheme (signature amortization, S-A) for BGP path authentication. Our simulation experiments conclude that the new approach has minimal impact on BGP convergence.

There are also other studies on more efficient mechanisms for securing BGP. One challenge in the adoption of any inter-domain routing security solution is its integration with existing infrastructure. In the *Inter-domain Routing Validation (IRV)* project [8], participating ASes host servers called IRVs. Each IRV maintains a consistent corpus of routing data received and advertised. Remote entities (e.g., routers, other IRVs, application) validate locally received data by querying source AS IRVs using an out-of-band (and potentially secure) protocol. This approach has the advantage that the query responses can be tailored to the requester for optimization or access control.

A recent effort that attacks the scalability issue of S-BGP is *psBGP* [40]. The major goal is to increase practicability of security solutions on BGP. The psBGP protocol contains four main components—authentication of AS numbers, authentication of IP prefix ownership, authentication of BGP speakers, and integrity of AS path. Essentially, this proposal combines aspects of S-BGP and soBGP.

Besides public key cryptography, there are efforts on securing BGP using symmetric key algorithms [9, 13, 42]. These proposals are more efficient on the operational latency, but require more storage, loose time synchronization, and complex key-pair pre-distribution.

Subramanian et al. [36] proposed the *Listen* and *Whisper* protocols to address the BGP security problem. The Listen protocol helps data forwarding by detecting "incomplete" TCP connection; the Whisper protocol uncovers invalid route announcements by detecting inconsistency among multiple update messages originating from a common AS. The Listen and Whisper approach dispenses with the requirement of PKI or a trusted centralized database, and aims for "significantly improved security" rather than "perfect security."

8 Conclusions

Implementation details of securing BGP have significant impact on BGP's behavior, and on the capacity of routers to actually use the algorithms. BGP's detailed time and memory consumption is too complex to analyze purely with mathematics, and so we turn to large-scale discrete-event simulation to examine the impacts of cryptographic operations and standard PKI certificate validation schemes on recent proposals to secure BGP.

We compare several major security proposals with S-BGP. Our simulation results have shown that it is possible to apply more efficient cryptographic operations to improve the performance in terms of convergence time, message size, or storage costs. Tradeoffs exist. Different proposals have their own strengthens and weakness. In particular, Signature Amortization achieves fast convergence at the cost of longer message size and more memory. Sequential Aggregation Signatures can decrease the message size, but slowing down the BGP convergence significantly. The Origin Authentication scheme can achieve instant origin proofs with in-band distribution of attestations, at the cost of exposing vulnerabilities to attackers.

We also analyzed the impacts of standard certificate revocation/validation mechanisms. The OCSP approach greatly slows down convergence. On the other hand, if BGP speakers rely on CRLs for certificate validation, the extra overheads by CRL handling operations are insignificant to affect convergence. Of course, such choices trade performance with security.

Besides BGP routing system, a variety of other large-scale distributed systems assume an underlying PKI—but neglect to consider its performance impact. Understanding the impact of the underlying PKI systems is a challenging task. In the future, we plan to analyze broader issues of PKI design and deployment that satisfy the security and performance requirements by these large-scale distributed systems and applications.

In ongoing work, we are also exploring new path authentication protocols that further improve performance.

Acknowledgments

The authors are grateful to Patrick McDaniel, Kevin Butler, William Aiello, Steve Kent, Scot Rea, B. J. Premore, and Hongsuda Tangmunarunkit for their valuable suggestions. This research has been supported in part by Sun, Cisco, the Mellon Foundation, NSF (CCR-0209144, EIA-9802068), AT&T/Internet2 and the Office for Domestic Preparedness, Department of Homeland Security (2000-DT-CX-K001). This paper does not necessarily reflect the views of the sponsors.

References

[1] William Aiello, John Ioannidis, and Patrick McDaniel. Origin Authentication in Interdomain Routing. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pages 165–178. ACM Press, October 2003.

[2] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. A Survey of Two Signature Aggregation Techniques. *RSA CryptoBytes*, 6(2):1–10, 2003.

[3] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In *Proceedings of Eurocrypt 2003*, number 2656 in LNCS, pages 416–432. Springer-Verlag, 2003.

[4] CIDR BGP Reports from AS1221 (Telstra), October 2004. `http://www.cidr-report.org/as1221/`.

[5] James Cowie, David Nicol, and Andy Ogielski. Modeling the Global Internet. *IEEE Computing in Science and Engineering*, 1(1):42–50, Jan.–Feb. 1999.

[6] Michalis Faloutsos, Petrof Faloutsos, and Christos Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proceedings of ACM SIGCOMM'99*, pages 251–262. ACM Press, 1999.

[7] Lixin Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking (TON)*, 9(6):733–745, December 2001.

[8] Goeffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, Patrick McDaniel, and Aviel Rubin. Working around BGP: An Incremental Approach to Improving Security and Accuracy in Interdomain Routing. In *The 10th Annual Network and Distributed System Security Symposium*, San Diego, California, February 2003.

[9] Michael Goodrich. Efficient and Secure Network Routing Algorithms. provisional patent filing, `http://www.cs.jhu.edu/~goodrich/cgc/pubs/rout-ing.pdf`, January 2001.

[10] Ramesh Govindan and Anoop Reddy. An Analysis of Internet Inter-Domain Topology and Route Stability. In *Proceedings of INFOCOM 1997*, pages 850–857, April 1997.

[11] Timothy G. Griffin and Gordon Wilfong. An Analysis of BGP Convergence Properties. In *Proceedings of SIGCOMM 1999*, pages 277–288, August 1999.

[12] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC3280, `http://www.ietf.org/rfc3280.txt`, April 2002.

[13] Yih-Chun Hu, Adrian Perrig, and Marvin Sirbu. SPV: Secure Path Vector Routing for Securing BGP. In *Proceedings of SIGCOMM 2004*, pages 179–192. ACM Press, August 2004.

[14] IANA: Internet Assigned Numbers Authority. `http://www.iana.org`.

[15] Internet corporation for assigned names and numbers. `http://www.icann.org`.

[16] Stephen Kent, Charles Lynn, Joanne Mikkelson, and Karen Seo. Secure Border Gateway Protocol (S-BGP) – Real World Performance and Deployment Issues. In *The 7th Annual Network and Distributed System Security Symposium (NDSS'00)*, San Diego, California, February 2000.

[17] Stephen Kent, Charles Lynn, and Karen Seo. Secure Border Gateway Protocol. *IEEE Journal of Selected Areas in Communications*, 18(4):582–592, April 2000.

[18] Steve Kent. Securing the Border Gateway Protocol: A Status Update. In *Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, October 2003.

[19] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet Routing Convergence. In *Proceedings of SIGCOMM 2000*, pages 175–187, August 2000.

[20] Craig Labovitz, Abha Ahuja, and Farnam Jahanian. Experimental Study of Internet Stability and Wide-Area Backbone Failures. In *Proceedings of the International Symposium on Fault-Tolerant Computing*, June 1999.

[21] Craig Labovitz, Abha Ahuja, Roger Wattenhofer, and Srinivasan Venkatachary. The Impact of Internet Policy and Topology on Delayed Routing Convergence. In *Proceedings of INFOCOM 2001*, pages 537–546, April 2001.

[22] Anna Lysyanskaya, Silvio Micali, Leonid Reyzin, and Hovav Shacham. Sequential Aggregate Signatures from Trapdoor Permutations. In *Eurocrypt 2004*, volume 3027 of *LNCS*, pages 74–90. Springer-Verlag, 2004.

[23] Zhuoqing Morley Mao, Ramesh Govindan, George Varghese, and Randy H. Katz. Route Flap Damping Exacerbates Internet Routing Convergence. In *Proceedings of SIGCOMM 2002*, August 2002.

[24] R. Merkle. Protocols for Public Key Cryptosystems. In *Proc 1980 Symposium on Security and Privacy, IEEE Computer Society*, pages 122–133, April 1980.

[25] R. Merkle. A Certified Digital Signature. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238. Springer-Verlag, 1990.

[26] S. Murphy. BGP Security Vulnerabilities Analysis. Internet-Draft, draft-murphy-bgp-vuln-00.txt, NAI Labs, February 2002.

[27] David M. Nicol, Sean W. Smith, and Meiyuan Zhao. Evaluation of Efficient Security for BGP Route Announcements using Parallel Simulation. *Simulation Practice and Theory Journal, special issue on Modeling and Simulation of Distributed Systems and Networks*, 12(3–4):187–216, July 2004.

[28] Andy T. Ogielski and James H. Cowie. SSFNet: Scalable Simulation Framework - Network Models. `http://www.ssfnet.org`. See `http://www.ssfnet.org/publications.html` for links to related publications.

[29] OpenSSL: The Open Source toolkit for SSL/TLS. `http://www.openssl.org`.

[30] Dan Pei, Xiaoliang Zhao, Lan Wang, Dan Massey, Allison Mankin, S. Felix Wu, and Lixia Zhang. Improving BGP Convergence through Consistency Assertions. In *Proceedings of INFOCOM 2002*, June 2002.

[31] Brian Premore. *An Analysis of Convergence Properties of the Border Gateway Protocol Using Discrete Event Simulation*. PhD thesis, Dartmouth College, June 2003.

[32] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC1771, `http://www.ietf.org/rfc1771.txt`, March 1995.

[33] The Route Views Project. `http://www.antc.uoregon.edu/route-views/`.

[34] Aman Shaikh, Anujan Varma, Lampros Kalampoukas, and Rohit Dube. Routing Stability in Congested Networks: Experimentation and Analysis. In *Proceedings of SIGCOMM 2000*, pages 163–174, August 2000.

[35] B. Smith and J.J. Garcia-Luna-Aceves. Efficient Security Mechanisms for the Border Gateway Routing Protocol. *Computer Communications (Elsevier)*, 21(3):203–210, 1998.

[36] Lakshminarayanan Subramanian, Volker Roth, Ion Stoica, Scott Shenker, and Randy H. Katz. Listen and Whisper: Security Mechanisms for BGP. In *Proceedings of First Symposium on Networked Systems Design and Implementation (NSDI 2004)*, March 2004.

[37] MitreTek Systems. Certificate Arbitrator Module. `http://cam.mitretek.org/cam/`.

[38] Hongsuda Tangmunarunkit, Ramesh Govindan, Scott Shenker, and Deborah Estrin. The Impact of Routing Policy on Internet Paths. In *Proceedings of INFOCOM 2001*, pages 736–742, April 2001.

[39] Iljitsch van Beijnum. *BGP: Building Reliable Networks with the Border Gateway Protocol*. O'Reilly, 2002.

[40] Tao Wan, Evangelos Kranakis, and P.C. van Oorschot. Pretty Secure BGP (psBGP). In *The 12th Annual Network and Distributed System Security Symposium (NDSS'05)*, San Diego, California, February 2005.

[41] Russ White. Architecture and Deployment Considerations for Secure Origin BGP (soBGP). IETF Internet Draft `http://www.ietf.org/internet-drafts/draft-white-sobgparchitecture-00.txt`, May 2004.

[42] K. Zhang. Efficient Protocols for Signing Routing Messages. In *The 5th Annual Network and Distributed Systems Security Symposium (NDSS'98)*, San Diego, California, March 1998.