The Performance Impact of BGP Security

Meiyuan Zhao and Sean W. Smith, Dartmouth College
David M. Nicol, University of Illinois at Urbana-Champaign

Abstract

The Border Gateway Protocol possesses security vulnerabilities, because speakers can lie. Proposed security protocols can address the vulnerabilities, but can significantly increase the performance overheads, which prevent their real-world deployment. Public key cryptography can address these vulnerabilities. We report our research into analyzing recently proposed security protocols and building simulation frameworks to evaluate their costs, and designing and validating techniques to reduce them.

The Border Gateway Protocol (BGP) is the standard way autonomous systems (ASes) within the Internet establish and maintain routing information between their domains. However, this protocol implicitly depends on hearsay, since each BGP *speaker* believes and repeats what it has heard from other speakers. This dependency introduces security problems, since a malicious speaker can forge claims that will then be propagated throughout the network, corrupting routing. There have been several proposed solutions to address these BGP vulnerabilities. A natural approach that provides strong security is to use *public key cryptography*: a speaker could verify the authenticity of each component of an announced route it receives, if each speaker along the way digitally signs the data it adds. However, using public key cryptography in this manner has costs: it takes time to generate and verify signatures, and check certificate status; and signature information makes announcements larger and increases storage requirements. Furthermore, evaluating how much these costs affect overall performance can be tricky. The complexity of the Internet makes an analytical approach difficult; the reality of Internet makes empirical approaches also unworkable.

As researchers with interests in both public key infrastructure (PKI) and networking, we have been exploring these issues. We identified recent proposals on securing BGP, and analyzed their performance and security trade-offs. We also set up a framework using large-scale network simulation to evaluate the costs of *S-BGP*, the primary PKI-based way to secure BGP. However, the more we looked at the performance, the more issues we saw — and the more potential ways to improve performance [1–3].

This article presents an overview of this work. We introduce BGP routing and major elements for a potential security solution. We review recent BGP security proposals. We present our evaluation frameworks and performance analysis results. We then conclude with some avenues for future work.

BGP and Security Solution Elements

BGP manages interdomain routing between ASes. ASes obtain IP address blocks for their networks from the IP address allocation hierarchy rooted at ICANN. Each allocation is a chain of entities starting at ICANN, followed by regional address space allocation authorities, Internet service providers (ISPs), data service providers (DSPs), and end users, each allocating subblocks of its address space to the next entity. In practice, BGP is currently the only routing protocol maintaining reachability between ASes. BGP thus plays a critical role in Internet efficiency, reliability, and security. For securing BGP itself, a solution may involve several players: ASes, BGP routers (speakers), the route registries that manage AS numbers and IP address allocation, and the relevant organizations (DSPs/ISPs). As BGP speakers maintain their routing tables by hearsay information, a security solution should protect the route announcements exchanged by speakers. Routing decisions by speakers mostly rely on two critical route attributes: *IP prefix* and *AS path*. Hence, security solutions typically focus on protecting these attributes from malicious attacks and misconfiguration. Before we discuss any specific security proposal for BGP, let us first review several building blocks that help construct a concrete security solution. Details of the cryptographic algorithms discussed in this section can be found in [4].

Hashing

Many security systems use *cryptographic hashing* for data integrity. This term is often reduced to just *hashing*, thus creating potential ambiguity with nonsecurity uses of the term. Standard hash functions such as MD5 and SHA-1 are computationally efficient and have historically provided nice *collision resistance*. Given a random value x, a one-way function $H(x)$ is collision-resistant if it is computationally infeasible to recover x given the value of $H(x)$ and to find $x' \neq x$ such that $H(x') = H(x)$. Recent cryptanalysis has suggested weaknesses in these standard functions (e.g., [5, 6]); the next few years should be interesting.

More advanced hashing techniques build on these basic hash functions. *One-way hash chains* are constructed by selecting a final value at random, and repeatedly applying H to derive *previous* values. The main property of the values of a one-way hash chain is that once Alice trusts the authenticity of a value y in the chain, she can derive trust on all previous values, but she is not able to compute the values following y. Thus, the party who constructed the chain can gradually release a sequence of authenticatable values.

A *Merkle hash tree* is another technique that uses hash func-

0890-8044/05/$20.00 © 2005 IEEE

tions to provide authentication of a set of messages. The hash values of the messages are the tree leaves. The tree is constructed by repeatedly applying H on concatenation of two hash values until constructing the root. The sender of a message m can prove its membership in the message set by providing a list of hash values, defined as the *hash path*, that help the receiver to reconstruct the root of the tree with m.

Signatures

Put simply, a cryptosystem consists of a way to transform data in a way that effectively hides information about the original data, unless one knows the key to transform it back. In *symmetric* cryptosystems (e.g., Data Encryption Standard, DES, or Advanced Encryption Standard, AES) the same key is used for both directions. In *asymmetric* cryptosystems, different keys are used — with the result that one can be made *public* while the other remains *private*. Hence, asymmetric systems are also known as *public key cryptography*.

The canonical textbook example of an asymmetric cryptosystem is *Rivest-Shamir-Adleman* (RSA). In RSA the central security parameter is the length of the *modulus*, a specially constructed large integer (typically, moduli are currently 1024 or 2048 bits long). The keys are a pair of *exponents* matched to the modulus. The cryptographic transformation consists of raising an nonnegative integer (smaller than the modulus) to an exponent and taking the remainder relative to the modulus.

Although asymmetric cryptosystems are typically much slower than symmetric ones, asymmetry permits new functionality such as *digital signatures*. Alice can use her private key to derive a *signature* on a message; anyone else can then use Alice's public key to verify that Alice indeed produced that signature from that message. In RSA-based signatures, the public exponent can be very small, making verification quick; however, the private exponent is long, increasing signature generation time. The signature length is the size of the modulus. Benchmarks show that 1024-bit RSA costs 10.0 ms for signing and 0.5 ms for verification on a 1 GHz processor [1].

DSA is an alternate public key signature scheme. The construction of DSA is more complex; verification tends to be more than twice as expensive as signing. Signatures are twice the length of the q modulus in DSA, totaling 320 bits.

With a secure storage area, the signer can *precompute* the message-independent values for future DSA signatures. This technique dramatically speeds up signing time. The benchmarks we obtained show that the running times of verification, signing, and signing with precomputation are 6.2 ms, 5.1 ms, and 3.0 μs, respectively [1].

Both RSA and DSA operate in a model where a signer generates one signature per message. Recently, *aggregate signature* schemes have emerged that permit many signers to combine their signatures into one data block, saving collective signature space. In the sequential aggregate signature (SAS) approach [7], a sequence of signers $s_1, \ldots, s_n$ can incrementally sign a sequence of messages ($m_1, \ldots, m_n$, respectively), producing an aggregate signature σ. SAS can be built from a variation of RSA. We estimate the running times of aggregate signing and verification operations as close to the running times by corresponding RSA signing and verification.

One-time signature schemes are another type of signature mechanism, in which a key cannot sign more than one message and remain secure. Unlike the above signature schemes, one-time signatures are generated using symmetric encryption and hashing. Typically, the signer generates a list of secret random strings to sign a message m. The signature is constructed such that the signer can release a subset of the secret values to help verify the signature. Signature generation and verification for one-time signatures are very efficient. The signature length depends on parameters and the message. The major drawback of one-time signatures is that if the same parameters are used to sign more than one message, the signatures can be forged.

Certificates

Effectively using digital signatures requires that the verifier knows the public key of the signer. Many BGP security solutions exploit standard *certificate* and PKI technology for this purpose. A certificate is a signed statement that binds an identity with its public key.

We can use such certificates to authenticate AS numbers and speakers. Certificates can also express authorization. For instance, an organization can use a certificate to express that a certain autonomous system is the authorized owner of an IP prefix. Typically, the certificate expresses this authority by binding the public key with the AS number and the IP addresses which this AS owns.

Although a certificate tends to be long-lived, it needs to be *revoked* if the authenticity or authority it represents is no longer valid. In one standard approach, the authority who issued certificates generates and signs a certificate revocation list (CRL) that contains the serial numbers of revoked certificates [1]. Any party that wishes to validate a certificate may download CRLs from publicly available sources to verify that a given certificate is not revoked. CRLs are updated periodically. In contrast, a more timely approach for validating certificate status is the Online Certificate Status Protocol (OCSP), where online responders answer certificate status queries in real time [1].

System Monitoring

Some security solutions use systematic monitoring to protect BGP against malicious or abnormal routing behaviors. For instance, ASes can set up dedicated servers that contain a database of route information and routing policy configuration. These servers may exchange and serve authenticated data using a new protocol other than BGP [9]. The major advantage is that the authentication function is separated from normal routing activities. As for the cryptographic approaches above, Update messages need to carry additional cryptographic data for route authentication.

As there are multiple databases storing security information for BGP, speakers may check redundant information for consistency. Inconsistent information should at least raise an alarm. Compared to cryptographic techniques, this approach may not provide strong enough protection on routing information.

Furthermore, we may exploit network engineering techniques to help authenticate BGP route announcements. For instance, upon receiving a route announcement, the speaker may try to connect to the destination expressed in the prefix to prove that it is reachable. Routes to unreachable prefixes should not be considered valid.

BGP Security Proposals

We analyze current major proposals for securing BGP routing. Since almost all current efforts address route announcements between ASes, we do not discuss protection on information exchanged within an AS. The security proposals we discuss in this section focus on two properties. *Origin authentication* establishes whether the AS originating a prefix is authorized to advertise it. *Path authentication* provides authenticity of the AS path in a route.

Secure BGP (S-BGP) — Kent *et al.* [10] proposed Secure BGP (S-BGP) for both origin authentication and path authentication. Possibly the most concrete BGP security proposal to date, the S-BGP solution consists of three major components: IPSec, PKIs, and attestations. IPSec is used to provide protection of BGP sessions.

There are two PKIs based on X.509v3 certificates. One is for authenticating AS numbers and BGP speakers; the other is for expressing IP address allocation by authorities. Both PKIs are strict hierarchies rooted at ICANN. The second tier are regional Internet registries (RIRs), responsible for assigning AS numbers and router identifiers. RIRs also allocate IP address blocks to a chain of entities, such as ISPs, DSPs, and subscribers. S-BGP uses certificates to express all of these assignments. While the PKI designs are strictly hierarchical, they reflect the existing routing infrastructure.

The third component consists of attestations. *Address attestations* (AAs) are for origin authentication, and *route attestations* (RAs) are for path authentication. These are built from DSA digital signatures. An IP address owner signs an AA to authenticate that a certain AS is authorized to originate route announcements for an IP prefix. The receiver validates this statement by verifying the AA as well as a list of IP address allocation certificates leading from ICANN to the proper subscriber. A BGP speaker generates RAs to authenticate the AS path in an outgoing route announcement. The speaker signs a message consisting of the prefix, the current AS path, and the AS number of the intended recipient. The speaker also needs to send all previous route attestations on earlier components of the AS path. The computational overhead of RAs is high, since a speaker signs the same route n times if it is sent to n peers, and the recipient verifies k signatures if the AS path is of length k. One proposed optimization is to use caching to avoid route revalidation. Such an approach trades space for speed.

Mainly three factors have kept S-BGP from wide deployment: high computational cost, expensive space costs, and difficulty of establishing centralized PKIs.

soBGP — soBGP(e.g., [11]) also aims to provide both origin authentication and path authentication. soBGP proposes using a decentralized "web-of-trust" model for AS number authentication, and a centralized hierarchical PKI for IP prefix ownership authentication. This latter hierarchy is very similar to the S-BGP IP address allocation PKI. In deployment, soBGP proposes to use dedicated internal servers with databases to distribute and validate all the related certificates. soBGP defines a new type of BGP message, the SECURITY message, to help transport required certificates.

For path authentication, soBGP achieves a slightly different goal. soBGP builds a topology map of the paths of the entire network. Each AS builds an *ASPolicyCert* that contains a list of its peers. Upon receiving a route announcement, the speaker verifies that the announced AS path does exist in the topology map.

This approach does not provide as strong protection as S-BGP. It is not able to catch an AS path falsification if the forged one is a valid path according to the topology map. Furthermore, building a topology map of the entire Internet actually counters the distributed nature of BGP.

Origin Authentication — The Origin Authentication (OA) scheme [12] revisits the IP address allocation PKI by S-BGP. S-BGP assumed that this information is established *out of band*: the relevant relying parties have learned and validated these certificates before the protocol even starts. Aiello *et al.* proposed various approaches sending origin authentication information *in line*, inside Update messages. OA uses origin attestation tags (OATs) to express IP address delegations by organizations. There are four major constructions: an organization can sign each address delegation separately; it can combine all delegations into one list and sign it once; it can break this long list into sublists, one for each delegated-to organization, and sign each separately; or it can construct a Merkle tree and sign the root. We denote these variations *OA-Simple*, *OA-List*, *OA-AS-List*, and *OA-Tree*, respectively.

psBGP — Pretty Secure BGP (psBGP) [13] provides origin and path authentication. Mainly, it attacks the practicality problem of S-BGP PKI design from a different perspective, proposing a decentralized trust model for IP address allocation. Each AS generates a prefix assertion list (PAL) of bindings between AS number and prefixes. The first binding is for itself. The rest of the bindings are for each of its peers. The assertion is *proper* if it is consistent with the assertion made by at least one of its peers. This design does require an assumption: no two ASes are in collusion. The main drawback of psBGP is that this model does not match the common practice in the real world, where subscribers obtain IP addresses hierarchically from higher-level ISPs or DSPs.

Interdomain Route Validation — Goodell *et al.* [9] proposed the *Interdoman Route Validation* (IRV) *service*, a separate protocol that combines features of S-BGP and the Internet routing registry. Here, validation information is not carried in Update messages. Instead, the separate IRV service maintains the validation information and supports queries. The database can store any type of information, such as static address allocation information and dynamic routing table information from local BGP speakers. IP address ownership is validated by investigating inconsistencies. Speakers detect forged AS paths by seeking confirmations from the IRV service in each AS along the path. To improve efficiency, the speaker may choose to query routes at random intervals or more frequently from ASes that are closer to the local AS. This performance optimization compromises security. Since speakers skip some of the route validations, forged AS paths can still be propagated by speakers, and databases supporting IRV services can be "poisoned."

Signature Amortization — Signature Amortization (S-A) [1] is a more efficient path authentication scheme we designed to improve time performance by S-BGP RAs. We notice that the majority of cryptographic operations for path authentication are signature verifications. Thus, we chose RSA as the digital signature algorithm. To compensate for the expensive private key operations, we use bit vectors and Merkle hash trees to get the most impact from each operation. S-A replaces distinct signatures on a route announcement with a single one by using a bit vector (1 b/peer) to indicate intended recipients. Furthermore, as BGP speakers use minimum route advertisement interval (MRAI) to specify the minimum amount of time a speaker must wait before sending out successive batches of Updates, BGP speakers queue pending Updates in output buffers. While messages are waiting in output buffers, S-A collects them, groups them using bit vectors, and constructs a Merkle hash tree of all message groups. The final attestation for an AS path consists of the signature on the tree root, the corresponding bit vector, and the hash path facilitating recalculation of the tree root. S-A speeds up path authentication by allocating more space in Update messages.

In recent work we extended S-A to aggregated path authentication (APA) schemes that also improve space efficiency [2].

Signing a message once for all receiving peers requires that the verifier be able to tell which peers were intended receivers.

Proposals	Origin authentication				Path authentication			
	Design	Security	Overhead		Design	Security	Overhead	
			Time	Space			Time	Space
S-BGP	Hierarchical PKI local memory	Strong	Low	High	Signatures in message	Strong	High	High
soBGP	Hierarchical PKI separate database	Strong	Low	Low	Topology map	Medium	Low	Low
psBGP	Distribute PALs local memory	Medium	Low	High	Signatures bit vector	Strong	Low	Very high
IRV	Separate IRV servers	Strong	Low	Low	Distributed database	Medium	High	Low
OA	Delegation OATs in message	Strong	Low	High	–	–	–	–
S-A	–	–	–	–	Signature bit vector hash tree	Strong	Low	Very high
APA	–	–	–	–	Aggregate signature, bit vector, hash tree	Strong	Low	Medium
SPV	–	–	–	–	Hash chain hash tree one-time signature	Medium	Low	Very high
Listen Whisper	–	–	–	–	Consistency check TCP flow	Low	Low	Low

Table 1. *Qualitative comparison of security proposals. The security and performance comparison are relative to S-BGP. Space overhead refers to additional costs on Update messages or local memory. Additional costs outside BGP speakers are not considered.*

The approach we considered includes a bit vector in each signature, and includes a map (from bits to peers) in the speaker's X.509 certificate, similar to ASPolicyCert by soBGP. Rather than transient session conditions, these certificates represent business relationships between peering ASes. An alternative is to include a list of ASes as intended recipients [14].

Secure Path Vector — Hu *et al.* [15] proposed the *secure path vector* (SPV) for BGP path authentication. Instead of expensive public key cryptography, SPV relies on symmetric cryptographic primitives: one-way hash chains, Merkle hash trees, and one-time signatures. The authors compared SPV with S-BGP (assuming S-BGP uses the RSA algorithm). SPV provides a similar level of security with an assumption: no neighboring ASes are colluding. Furthermore, one-time signatures are secure when they are used to sign only one message. Multiple Updates for the same prefix can increase the probability of falsifying AS paths. In terms of performance, the authors concluded that SPV performed 22 times faster with 2.7 times longer signatures.

Listen and Whisper — Subramanian *et al.* proposed the *Listen* and *Whisper* protocols for efficient path authentication [16]. Similar to psBGP and IRV, speakers rely on consistency checks to detect invalid route information. The Whisper protocol uses hash function and redundant routes to detect inconsistency on AS paths. One detection triggers an alarm. Multiple alarms can help identify a malicious AS. Furthermore, the Listen protocol uses TCP flows to verify that a specific prefix is actually reachable. Compared to other security proposals, Listen and Whisper offers weaker security. However, it does not rely on any centralized infrastructure or database.

Performance Evaluation

Analysis

The above security proposals have their strengths and drawbacks. We are interested in understanding their performance impacts, one important factor that affects their deployment in the real world. Table 1 summarizes their properties and compares their performance qualitatively, based on the performance evaluation in studies by authors of the proposals.

Most of the security proposals use hierarchical PKI architecture to express and verify IP address ownership. Generally, this approach provides relatively stronger security than distributed models based on consistency checking. Most of the proposals choose to prepare address ownership proofs offline. The time overhead of sending and verifying a proof is fairly low. There are two types of approaches to store these proofs. soBGP and IRV use independent servers and/or even independent supporting protocols to manage the databases of certificates and attestations. This approach offloads the memory burden from BGP speakers. Moreover, independent supporting protocols allow servers to distribute and validate the database in the background.

Approaches to path authentication demonstrate diversity. Digital signature approaches provide strong security. S-BGP incurs significant overhead costs in computation and space. S-A and psBGP use additional tricks to speed up signing and verification operations. Unfortunately, such techniques result in even higher space costs. Aggregated path authentication (APA) schemes are the best of all signature-based approaches in terms of time and space costs. Other proposals use approaches other than digital signatures to improve performance. SPV uses symmetric cryptographic primitives. soBGP, IRV, and Listen and Whisper apply a monitoring approach. However, all of them degrade security to some degree. Inter-

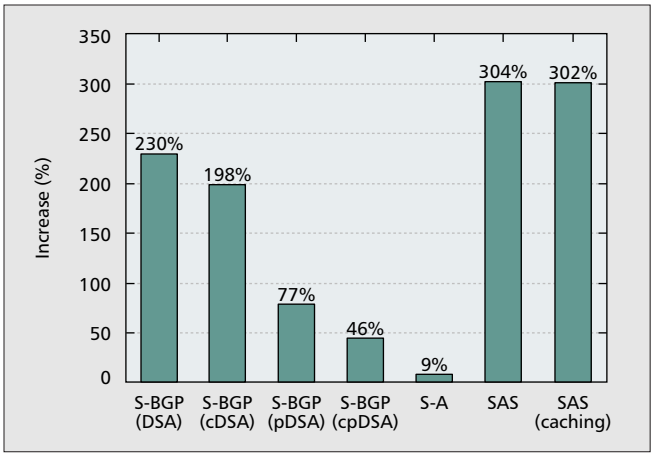

Figure 1. *Relative increase in convergence time of path authentication schemes relative to ordinary BGP.*

estingly, although SPV uses symmetric cryptographic primitives to speed up Update message processing, it actually increases space costs. Database approaches such as soBGP and IRV use independent servers to distribute path authentication information. Such approaches reduce the space costs because speakers do not process or store security-related data locally. However, their protection of AS paths is not very satisfying. There always seem to be trade-offs between performance and security.

With this preliminary comparison, no single proposal stands out as the best solution to BGP security. One might argue that security and performance are not the only two factors that affect the practicability of a security proposal; flexibility and scalability are important design criteria as well. We are confident that it is very difficult for a security proposal with expensive cost to achieve other nice properties at the same time. Next, we use simulation to obtain more concrete understanding of performance impacts.

Simulation Evaluation

We use SSFNet to simulate performance of some of these BGP security proposals. The simulation model contains an AS-level topology of 110 ASes and 110 BGP speakers. Our previous reports [1–3] present the details of the complete evaluation framework for performance study. The simulated network has one BGP speaker for each AS. Although simplified, single-speaker ASes suffice for this performance study because the security protocols are mostly focused on inter-AS BGP sessions. We provided detailed discussion on model validation in [1].

The experiments measure performance during router reboot. The goal is to understand performance impacts while BGP speakers are under stress. We use three major measurement metrics: BGP convergence time, message size, and memory costs. We present mean values of 20 simulation runs. Convergence time is the most important metric we use to understand computational overheads by security protocols. We choose convergence time rather than CPU utilization because it better reflects the impact on BGP behavior of the security protocols.

A routing systems is said to be unstable if it converges slowly. BGP instability causes all sorts of network problems. To avoid unexpected effects of network traffic on BGP behavior, we choose not to model abnormal network activities such as dropping packets or congestion. Certainly, as we develop more concrete simulation models for BGP security, it would be very useful to examine what impacts the prolonged convergence time may have on packet-level network traffic.

Path Authentication — We focus our evaluation on proposals that use digital signatures for strong authentication on AS paths. We simulated S-BGP with standard DSA. We also simulated S-BGP with several unimplemented optimizations:

- With caching of generated and verified signatures (cDSA)
- With precomputation of all DSA signatures (pDSA)
- With caching and precomputation (cpDSA)

We also simulated S-A and S-A with validated path caching. Finally, we applied the SAS technique to S-BGP route attestations to decrease space cost. We simulated this S-BGP variant as well as the SAS caching validated signatures.

Convergence Time — Figure 1 shows the relative impact of security protocols on convergence time. We start with a baseline of 153.7 s for ordinary BGP without any security extension. All the variants of S-BGP considered significantly increase convergence time; even optimized, convergence time is 46 percent larger than ordinary BGP. Such slowdowns can 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. On the other hand, S-A achieves much faster convergence without the additional burden of caching large amounts of data in memory. Interestingly, the effort of applying SAS to S-BGP exacerbates slow convergence, mainly because like RSA, private key operations by SAS are expensive compared to DSA.

In simulations we modeled running times based on standard crypto libraries on typical processors. The application of hardware accelerators will speed up security operations. For a fixed problem size, this will reduce the relative impact of cryptography on convergence. However, as we increase the size of the networks, the number of announcements processed during rebooting grows superlinearly, and the differences between protocols again becomes significant. From the point of view of scaling, everything that can be done to reduce overhead must be done to reduce it, including hardware acceleration and software optimizations.

Message Size — SAS produces one signature for an AS path; it outperforms other schemes on message size. S-A carries long signatures, bit vectors, and hash paths for an AS path. The resulting Update messages are extremely long. Our experiment results, shown in Table 2, present the difference. For both S-BGP and S-A, the number of signatures in messages grows as the path length increases.

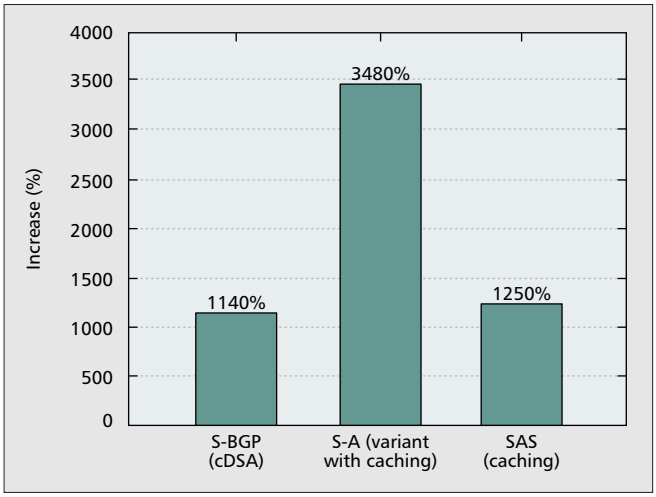

Figure 2. *Relative increase in memory costs of path authentication schemes relative to ordinary BGP.*

	BGP	S-BGP	S-A	SAS	S-A with SAS
Average message size (bytes)	36.09	318.61	1107.08	184.29	638
Increase		8.83	21.57	5.11	17.7
Maximum message size (bytes)	42.6	527.7	1915.4	191.2	1138
Increase		13.77	47.75	4.40	26.7

Table 2. *Message size. The increase numbers are based on the message size by original BGP.*

Experiments show that SAS is an attractive approach for reducing message size, so it was natural to combine with S-A. This new variant mitigates the increased message size to some degree. However, the message still needs to carry bit vectors and hash paths.

Memory Costs — Figure 2 shows the average memory cost for individual BGP speakers. We start with a baseline of 9 kbytes memory at each speaker for ordinary BGP. On average, S-BGP costs 11.4 times more memory to cache routes. Memory cost of S-A is significantly higher because of RSA signature size, additional bit vectors, and hash tree paths. Although there is only one aggregate signature for each AS path, SAS still costs as much memory as S-BGP. Thus, simply applying SAS to S-BGP does not help improve efficiency in terms of either time or space. In our latest study we examined aggregate signature techniques thoroughly and designed new aggregated path authentication schemes that outperform S-BGP in both time and space [2].

Origin Authentication — Our simulations concentrate on the OA schemes that allow inline propagation of IP address ownership proofs. As only verifications are involved for ordinary Update processing, experiments showed that there are relatively insignificant increases in convergence time. However, all the approaches showed an increase in memory costs and message size, particularly the OA-List approach. The OA-AS-List approach is most efficient, but also the closest to the original S-BGP method of origin authentication.

As we have seen, most origin authentication schemes adopt similar forms of expressing IP address ownership in certificates. The criterion that differentiates them is how they distribute and use certificates. Whether they use a local cache or an independent database, the resulting space costs are similar. Kent analyzed that the scale of the Internet in 2003 required about 75–85 Mbytes space to store all certificates [14]. This number is applicable to other origin authentication schemes.

Certificate Validity — Many security proposals ignore the issue of certificate validity, despite the security of their schemes relying heavily on certificate validity. Although certificate revocations do not happen as frequently as BGP Update messages, we should not ignore their impacts.

In simulation we assumed that BGP speakers validate certificate status before they use the corresponding public key to verify IP address ownership proofs or to verify signatures on routes, and then studied the performance impacts of certificate revocation on S-BGP. The simulation model assumes that each organization or autonomous system has a repository that offers certificates and CRLs and an online OCSP responder that can answer queries about certificate status. We simplify the task of certificate validation to just checking certificate status.

First, we examine the performance for S-BGP origin authentication. We used the approximated IP address allocation hierarchy for the Internet by Aiello *et al.* [12] for experiments. Organizations prove their IP address ownership using a list of certificates that connects the prefix to the root. Experimental results showed that using OCSP to check certificate status for each prefix is prohibitively expensive: it costs a BGP speaker five times more time to converge. On the other hand, using CRLs is manageable since it is less aggressive: BGP speakers only download necessary CRLs when they do not have a fresh local copy. A few missing copies slow down convergence only slightly. Then for S-BGP path authentication, the results are similar. BGP speakers verify status of certificates for BGP speakers. Our simulation showed that using OCSP for verification checking in S-BGP — even using OCSP with parallel requests — was a bad idea. Using CRLs was more acceptable, and only showed a linear slowdown in convergence time as the number of expired CRLs increased.

Conclusions and Future Work

A number of proposals have been made to secure BGP. However, their performance has significant impact on BGP's behavior and the capacity of routers to actually use the protocols. In our research we have studied most of the recently proposed BGP security protocols and examined their performance issues. Our analysis has shown that there are trade-offs between security and performance. Strong security incurs costs.

BGP's detailed time and memory consumption is too complex to analyze purely with mathematics, so we turn to large-scale network simulation. Our simulation results have shown that it is possible to apply more efficient cryptographic operations to improve performance in terms of convergence time, message size, or storage costs. We explored the trade-offs between fast convergence by Signature Amortization and space efficiency by Sequential Aggregated Signature. These results have led us to design aggregated path authentication schemes that are efficient in both time and space in recent work.

Our simulation model has limitations. Currently the model is focused mostly on S-BGP with various signature schemes. Besides signature-based schemes, there are a number of proposals that use database and other techniques intensively. For concrete performance examination, we plan to extend our simulation to model these approaches. The model will further explore techniques to save space on caching routes and related security data. The latest S-BGP proposal now requires that routes that have been accepted, but whose signing certificates have now expired, be considered withdrawn; this modification suggests further avenues for simulation. 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 paths for them. This also is an avenue for future work.

Acknowledgments

The authors are grateful to Steve Kent, Patrick McDaniel, Kevin Butler, William Aiello, Dan Boneh, Russ Housley, Scot Rea, B. J. Premore, Hongsuda Tangmunarunkit, and anonymous reviewers 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 article does not necessarily reflect the views of the sponsors.

A preliminary version of some of this material appeared in a conference paper [3].

References

[1] D. M. Nicol, S. W. Smith, and M. Zhao, "Evaluation of Efficient Security for BGP Route Announcements using Parallel Simulation," *Simulation Practice and Theory J.*, Special Issue on Modeling and Simulation of Distributed Systems and Networks, vol. 12, no. 3–4, July 2004, pp. 187–216.
[2] M. Zhao, S. W. Smith, and D. M. Nicol, "Aggregated Path Authentication for Efficient BGP Security," *Proc. 12th ACM Conf. Comp. Commun. Sec.*, Nov. 2005.
[3] M. Zhao, S. W. Smith, and D. M. Nicol, "Evaluating the Performance Impact of PKI on BGP Security," *4th Annual PKI R&D Wksp.*, Apr. 2005.
[4] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 2001.
[5] X. Wang, Y. Yin, and H. Yu, "Finding Collisions in the Full SHA-1," *25th Annual Int'l. Cryptology Conf.*, Aug. 2005.
[6] X. Wang and H. Yu, "How to Break MD5 and Other Hash Functions," *Eurocrypt 2005*, May 2005.
[7] A. Lysyanskaya *et al.*, "Sequential Aggregate Signatures from Trapdoor Permutations," *Eurocrypt 2004*, vol. 3027, *LNCS*, Springer-Verlag, 2004, pp. 74–90.
[8] C. Adams and S. Lloyd, *Understanding PKI*, Ch. 8, Addison-Wesley, 2nd ed., 2002.
[9] G. Goodell *et al.*, "Working around BGP: An Incremental Approach to Improving Security and Accuracy in Interdomain Routing," *10th Annual Net. and Distrib. Sys. Sec. Symp.*, San Diego, CA, Feb. 2003.
[10] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol," *IEEE JSAC*, vol. 18, no. 4, Apr. 2000, pp. 582–92.
[11] R. White, "Securing BGP Through Secure Origin BGP," *IP J.*, vol. 6, no. 3, Sept. 2003, pp. 15–22.
[12] W. Aiello, J. Ioannidis, and P. McDaniel, "Origin Authentication in Interdomain Routing," *Proc. 10th ACM Conf. Comp. and Commun. Sec.*, Oct. 2003, pp. 165–78.
[13] T. Wan, E. Kranakis, and P. C. van Oorschot, "Pretty Secure BGP (psBGP)," *12th Annual Net. and Distrib. Sys. Sec. Symp.* San Diego, CA, Feb. 2005.
[14] S. Kent, "Securing the Border Gateway Protocol: A Status Update," *7th IFIP TC-6 TC-11 Conf. Commun. and Multimedia Sec.*, Oct. 2003.
[15] Y.-C. Hu, A. Perrig, and M. Sirbu, "SPV: Secure Path Vector Routing for Securing BGP," *Proc. SIGCOMM 2004*, Aug. 2004, pp. 179–92.
[16] L. Subramanian *et al.*, "Listen and Whisper: Security Mechanisms for BGP," *Proc. 1st Symp. Networked Sys. Design and Implementation*, Mar. 2004.

Biographies

MEIYUAN ZHAO (zhaom@cs.dartmouth.edu) is with Intel Technology and Research. She received her Ph.D. degree from the Department of Computer Science at Dartmouth College. She received her B.E. degree in computer science and engineering from the University of Electronic Science and Technology of China. Her research interests are in network security, routing, PKI, and simulation and modeling.

SEAN W. SMITH is on the faculty of the Department of Computer Science at Dartmouth College. His current research focuses on how to build trustworthy systems in the real world. He previously worked as a scientist at IBM T. J. Watson Research Center, doing secure coprocessor design, implementation, and validation; and at Los Alamos National Laboratory, doing security designs and analyses for a wide range of public sector clients. He was educated at Princeton (B.A., mathematics) and Carnegie Mellon University (M.S. and Ph.D., computer science).

DAVID M. NICOL [F] is a professor of electrical and computer engineering at the University of Illinois, Urbana-Champaign, and a member of the Coordinated Sciences Laboratory. He has a B.A. in mathematics from Carleton College (1979), and M.S. (1983) and Ph.D. (1985) degrees in computer science from the University of Virginia. His research interests are in high-performance computing, performance analysis, simulation and modeling, and network security.