Securing BGP Through Secure Origin BGP

by Russ White, Cisco Systems

Networks have come under increasing scrutiny in the area of security. Routing, the part of the network that provides information on how to reach destinations within the network, has been gaining attention from a security perspective as well. *The Internet Engineering Task Force* (IETF) has, in fact, formed a new working group, the *Routing Protocols Security Requirements Working Group* (**`http://www.rpsec.org`**), to analyze security in routing systems.

Of course, the biggest network in existence is the Internet, and the routing protocol that provides reachability and path information for the Internet is the *Border Gateway Protocol* (BGP), specified in RFC 1771. Several methods of securing the information carried within BGP have been proposed:

- *Internet Route Verification* (IRV), described in "Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing," Symposium on Network and Distributed Systems Security, February 2003, by Geoffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, Patrick McDaniel, and Aviel Rubin. IRV relies on out-of-band communication with a route originator to verify the correctness of a route.
- S-BGP, described in the companion article and at: **`www.net-tech.bbn.com/projects/s-bgp`**
- *Domain Name System* (DNS)-based *Network Layer Reachability Information* (NLRI) origin *Autonomous System* (AS) verification in BGP, which is the oldest attempt at validating the information carried within BGP, is described in **`draft-bates-bgp4-nlri-orig-verif-00.html`**,

This article discusses *Secure Origin BGP* (soBGP), a solution recently proposed by a group (including me) mostly within Cisco Systems. We believe soBGP to be a deployable mechanism for validating the correctness and authorization of the data carried within BGP, and also for preventing the sorts of attacks resulting from misconfiguration or intentional insertion of bad data into the Internet routing system.

We address four goals when we consider security in terms of BGP:

- Is the AS originating the destination (prefix) authorized to advertise it? In other words, if a router receives an advertisement for the 10.1.1.0/24 network originating in AS65500, is there any way to verify that AS65500 is supposed to be advertising 10.1.1.0/24?
- Does the AS advertising the destination actually have a path to the destination? In other words, if a router is receiving an advertisement from a BGP peer in AS65501 that it can reach 10.1.1.0/24, is there any way to verify that AS65501 actually has a path to the AS origination 10.1.1.0/24?

- Is the peer advertising the route authorized by the originator, or owner, of the destination, to advertise a path to the destination?
- Does the path advertised by a peer AS fall within the policies the local network administrators have set forward? The most obvious issue is whether or not the AS Path advertised by the peer is an acceptable path to send the traffic along.

We argue elsewhere that the second two goals cannot be fully met within an operational internetwork, for many reasons; see **`draft-white-pathconsiderations-00.txt`** for further discussion on this point. In this article, then, we discuss how soBGP can meet the first two goals in operational networks.

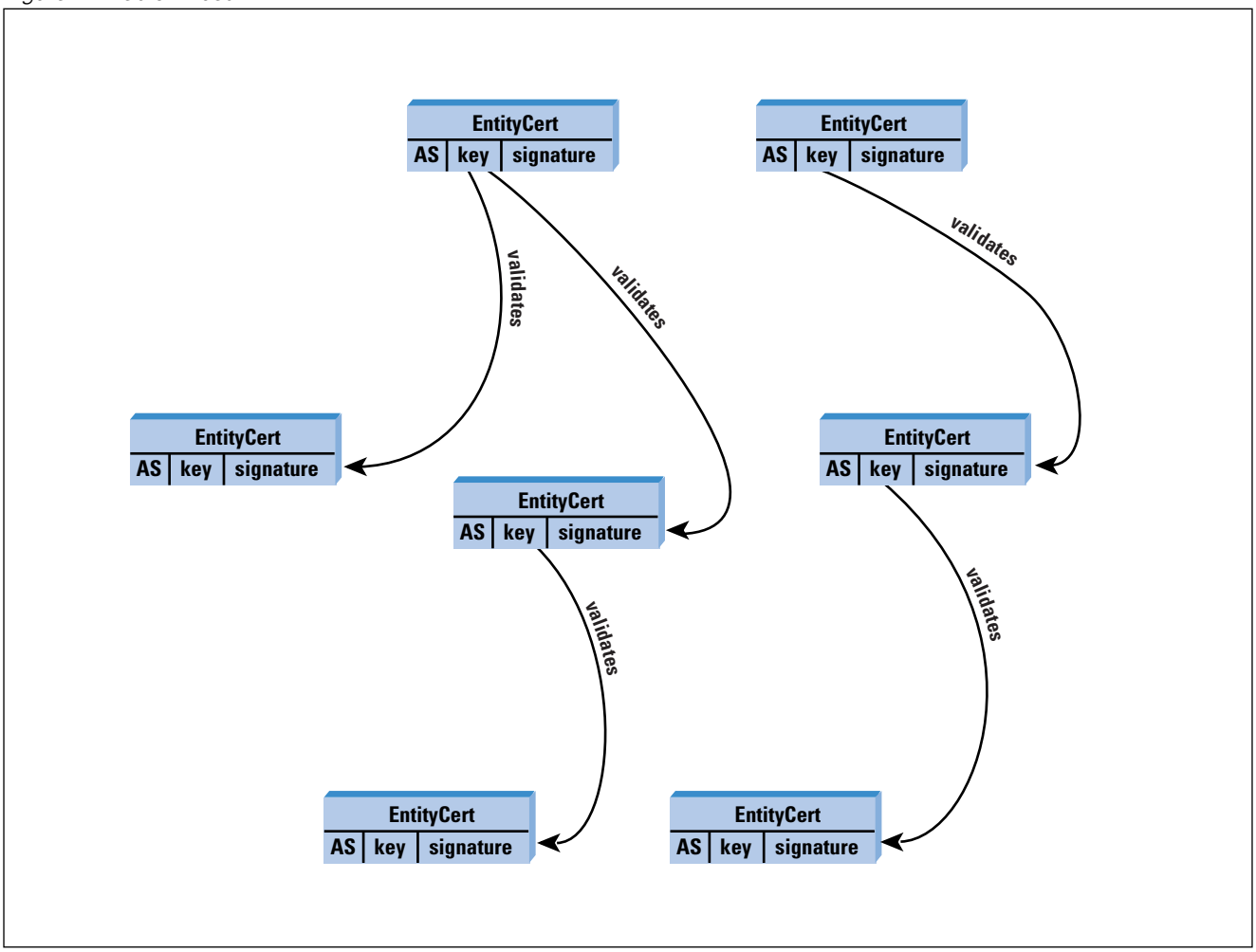
Begin at the Beginning: Who Are You?

The first step in securing anything is authentication; each participant must have some way of knowing who the other participants are, and what information they will be using to sign or encrypt their data. This is a classic problem in cryptography, called *key distribution*. There must be some way to receive keys used to sign or encrypt data, and then to validate that the keys received actually belong to the participant we believe they belong to.

This problem is addressed in soBGP using an *EntityCert,* which ties an AS number to a public key (or a set of public keys) corresponding to a private key the AS will be using to sign various other certificates. An EntityCert is defined in soBGP to be an X.509v3 certificate, similar to those used by *Transport Layer Security* (TLS) and *IP Security* (IPSec). The main problem we face when accepting an EntityCert is knowing whether or not the key carried within the certificate is actually the key of the advertising AS.

soBGP resolves this by requiring the EntityCert to be signed by a third party, validating that this AS actually belongs with this key. A small number of "root keys" distributed out of band could then be used to validate a set of advertised EntityCerts. These are used in turn to build up the database of known good ASm/key pairs in the system, allowing even more EntityCerts to be validated. Thus, EntityCerts can form a web of trust, built on the public keys of a small number of well-known entities, such as top-level backbone service providers, key authentication service providers (such as Verisign), and others.

Figure 1: Web of Trust

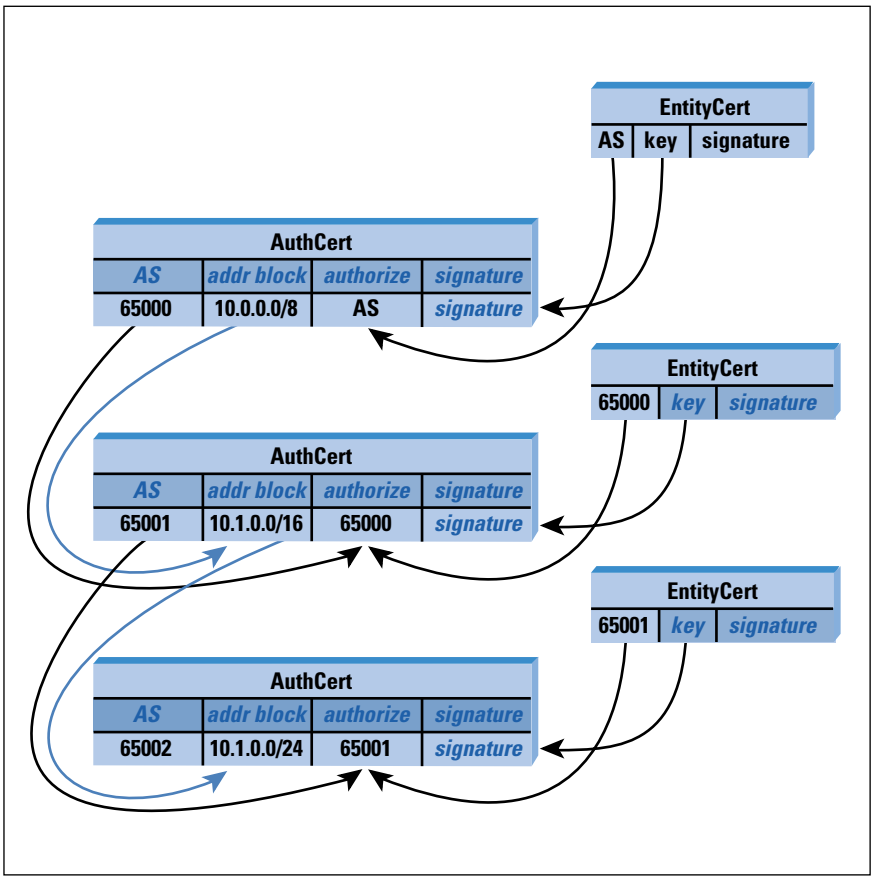


The key each AS distributes in its EntityCert is actually the public half of a private/public key pair. An AS would keep its private key entirely private, holding it on one highly secure device in its network (which is not even required to be online), and generating signatures for other certificates as needed. Only an AS public key is ever exposed in this way, so no special protection mechanisms (for example, tamper-resistant hardware) are required at any border to prevent private keys from being compromised.

The First Goal: Are You Authorized?

Now that we have distributed a public key per AS, we can build a certificate that will provide authorization for an AS to advertise a specific block of addresses. This authorization is provided through an *Authorization Certificate,* or *AuthCert*. An AuthCert ties an AS to a block of addresses that the AS may advertise, as Figure 2 illustrates.

Figure 2: Authorization Example



Starting at the top of the illustration, we find that some AS has authorized AS65000 to advertise prefixes within the block 10.0.0.0/8. The AuthCert is signed using the authorizing AS key. To delegate some part of this block of address space to another AS, AS65001, AS65000 builds an AuthCert tying 10.1.0.0/16 to AS65001. AS65001, in turn, suballocates a smaller part of this address space to AS65002, by building an AuthCert tying AS65002 to 10.1.1.0/24.

Any device receiving these three AuthCerts can check them by:

- Looking up the public key of the authorizer, and verifying the signature on the AuthCert
- Making certain the authorizer is permitted to advertise the address space it has suballocated this block of address space from

The device then builds a local table of address blocks and corresponding ASs authorized to advertise prefixes within those address blocks. Received updates can be checked against this database to verify authorization of the originating AS to advertise a prefix.

Blocks of address space are used here, rather than individual prefixes; an AuthCert can authorize an AS to advertise any number of prefixes within a block of addresses. This reduces the number of certificates within the system, thereby reducing overall cryptographic processing requirements. If a specific AS desires per-prefix authorization, it can build individual AuthCerts for each allocated prefix, rather than for blocks of address space.

Per-Prefix Policy

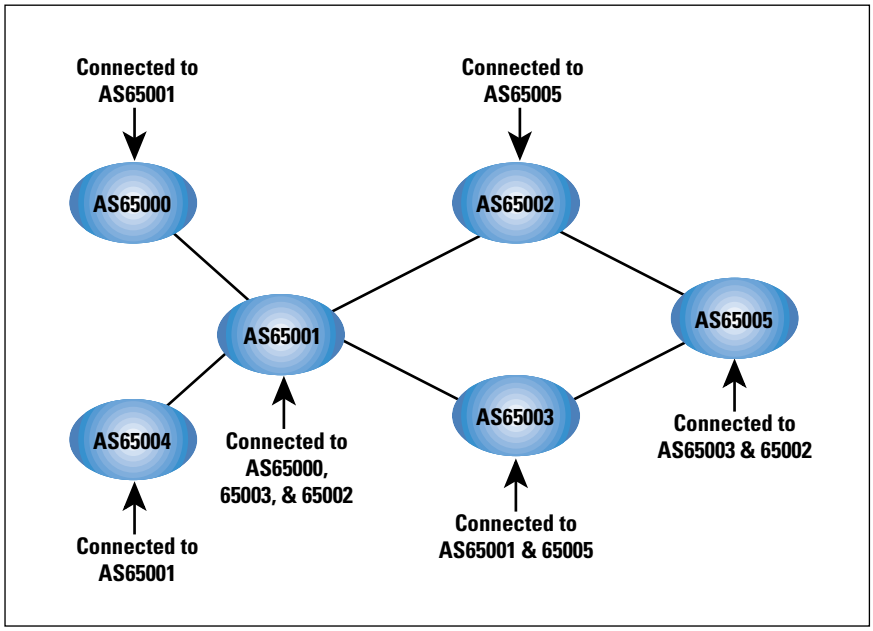
AuthCerts are not advertised as independent certificates within soBGP; instead, they are wrapped in a *PrefixPolicyCert*. PrefixPolicyCerts contain an AuthCert, a set of policies the originator would like to apply to prefixes advertised within this block of addresses, and a signature generated using the private key of the authorized AS. Policies that may be included in the PrefixPolicyCert include the longest prefix length allowed within the address block, and possibly other policies, such as a list of ASs that may not be or must be in the AS Path of routes to destinations within the address block.

In reality, the per-prefix policies available to the originator are limitless; the main problem is enforcing those policies when they are received by other ASs.

The Second Goal: Do You Really Have a Path?

Our second goal is to be able to verify that the advertiser of a given route actually has a path to the destination. This goal is met in soBGP by building a topology map of the paths of the entire internetwork. Each AS attached to the internetwork builds an *ASPolicyCert,* which contains, primarily, a list of its peers, and signed using the originator's private key. Using this list of transit peers, a map of the internetwork topology may be built, as Figure 3 illustrates.

Figure 3: Connectivity Graph Example



If AS65005 receives an update from AS65002, claiming it can reach a destination in AS65000 through the path {65002, 65001, 65000}, it can:

- Check to make certain AS65002 claims to be connected to AS65001 in its ASPolicyCert, and that AS65001 claims to be connected to AS65002 in its ASPolicyCert
- Check to make certain AS65001 claims to be connected to AS65000 in its ASPolicyCert, and that AS65000 claims to be connected to AS65001 in its ASPolicyCert

If, for instance, AS65002 claims a path to a destination inside AS65000 through the path (65002, 65000), AS65002 would be able to discover that the path is invalid, because AS65000 does not claim to be connected to AS65002. This simple two-way connectivity check along a graph can be mixed with various policy statements—stating a specific peer is not a transit, not advertising certain peers, etc.—to provide a much wider range of policies than AS Path-based methods.

Transporting Certificates

One of the primary problems any security system such as soBGP is going to face is transporting security information through the internetwork. We would like to make certain we do not rely on the routing system to provide information about the security of the routing system. In other words, we would not like to rely on unsecured routing information in order to reach a server providing the information required to secure the path to the server itself.

soBGP resolves this by proposing to advertise certificates in much the same way as routing information is propagated today—through an interdomain protocol. Currently the soBGP drafts specify a new type of BGP message, the SECURITY message, which can be used to transport the required certificates, the EntityCert, the PrefixPolicyCert, and the ASPolicyCert, throughout an internetwork. Other methods of transporting data such as these certificates throughout an internetwork are currently being pursued by the IETF; if other methods are offered, soBGP could transport certificates across any such distribution mechanism.

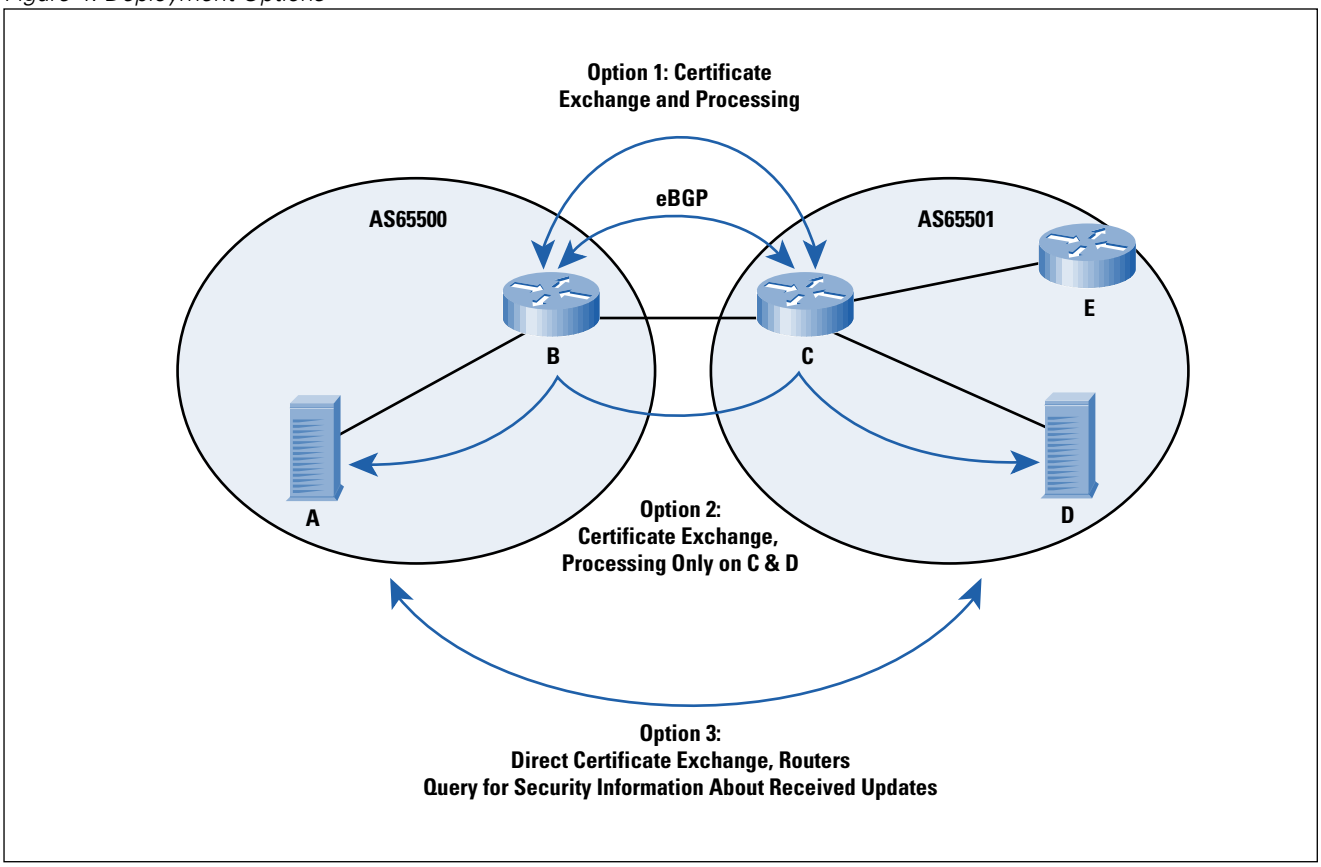
Deployment

Finally, we come to the hardest problem any routing security system is going to face: actually getting it operating in the field, with useful results, with a minimum of equipment changes, and a minimum number of participants. Here, soBGP provides a wide variety of options, primarily because it is not transport-dependent, nor dependent on a yet-to-be constructed centralized set of servers.

Although deployment options abound, here we discuss three, just to show the range of options available. Figure 4 illustrates these options.

The first option shown in this network is direct certificate exchange and processing between border routers. With this option, routers that are capable of the cryptographic processing required to validate received certificates exchange certificates with their peers in other ASs (just as they exchange routing information today), process those certificates, and build local databases from which they perform security checks on received updates.

Figure 4: Deployment Options



Although it may appear that processing, in this situation, would be extensive, it is actually possible to spread the processing required out among the border routers in a large AS. For instance, each certificate that router C receives and processes can be subsequently sent over an encrypted link to Router E. Router E could treat these certificates as though they had been validated locally, because they are received across an encrypted link from a trusted peer within the same administrative domain. Thus, only the edge router that has learned a certificate would actually process the certificate. This spreads the processing along all the edges in the AS.

A second option is for the edge routers, B and C, to exchange the certificates, but not process them. Instead, each edge router would relay the not-yet-validated certificates to internal servers A and D, respectively, thereby validating the certificates by performing the necessary cryptographic operations. As the border routers receive updates, they can query the server about the validity of each update, and take action based on the reply received.

Finally, it is possible for the servers to exchange certificates directly, over a multihop session. Servers A and D would then process the certificates, and the border routers, B and C, would query these servers to determine if received updates are valid or invalid.

Summary

Through this short survey of soBGP, we have shown it to be a flexible, moderately lightweight, yet strong system for validating the information carried through BGP in a large internetwork. It has low overhead processing requirements and very flexible deployment options, but no reliance on centralized servers. We are currently working to develop prototypes of soBGP on several platforms, to show how the technology will work on a wide range of devices.

For more information on soBGP, refer to:
`ftp://ftp-eng.cisco.com/sobgp/index.html`

You will find the most recent versions of the drafts, several slide shows, and other information about soBGP at this site.

RUSS WHITE works for Cisco Systems in the Routing Protocols Deployment and Architecture (DNA) team in Research Triangle Park, North Carolina. He has worked in the Cisco Technical Assistance Center (TAC) and Escalation Team in the past, has coauthored several books on routing protocols, including *Advanced IP Network Design, IS–IS for IP Networks,* and *Inside Cisco IOS® Software Architecture*. He is currently in the process of publishing a book on BGP deployment, and is the cochair of the Routing Protocols Security Working Group within the IETF. E-mail: **`riw@cisco.com`**