On Inter-domain Routing Security and Pretty Secure BGP (psBGP)

Evangelos Kranakis, P.C. van Oorschot, Tao Wan
Carleton University, Ottawa, Canada
{*kranakis, paulv, twan*}*@scs.carleton.ca*

It is well known that the Border Gateway Protocol (BGP), the IETF standard inter-domain routing protocol, is vulnerable to a variety of attacks, and that a single misconfigured or malicious BGP speaker could result in large scale service disruption. In this paper, we present *Pretty Secure BGP (psBGP)* – a proposal for securing BGP, including an architectural overview, design details for significant aspects, and preliminary security and operational analysis. psBGP differs from other security proposals (e.g., S-BGP and soBGP) in that it makes use of a single-level PKI for AS number authentication, a decentralized trust model for verifying the propriety of IP prefix origin, and a rating-based stepwise approach for AS_PATH (integrity) verification. psBGP trades off the strong security guarantees of S-BGP for presumed-simpler operation, e.g., using a PKI with a simple structure, with a small number of certificate types, and of manageable size. psBGP is designed to successfully defend against various (non-malicious and malicious) threats from uncoordinated BGP speakers, and can be incrementally deployed with some incremental benefits.

Categories and Subject Descriptors: C.2.6 [**Computer-Communication Networks**]: Internetworking—*Security*

General Terms: Inter-domain Routing, Security

Additional Key Words and Phrases: BGP, Trust, Routing Security, Secure Routing Protocols

1. INTRODUCTION AND MOTIVATION

The Internet routing infrastructure consists of a number of Autonomous Systems (ASes), each of which consists of a number of routers under a single technical administration (e.g., sharing the same routing policy). The Border Gateway Protocol (BGP) [Rekhter and Li 1995] is the IETF standard inter-domain routing protocol for exchanging reachability information between ASes on the Internet. Each network layer destination is identified by an IP prefix representing a range of IP addresses. An AS announces its IP prefixes via BGP to its direct neighbors, which may further propagate the prefix announcement to their neighbors. A remote AS receiving such announcement may build routes for forwarding traffic destined to the addresses within the address range specified by the announced prefixes.

One critical question with BGP is the following: which AS has a right to announce a given IP prefix? The current version of BGP does not have any mechanism to verify the propriety of IP prefix announcements. This opens a serious security hole which allows one AS to announce IP prefixes allocated or delegated (hereafter *assigned*) to any other ASes. This is commonly referred to as *prefix hijacking*. Examples of consequences include denial of service (i.e., legitimate user traffic cannot get to its ultimate destination) and man-in-the-middle attacks (i.e., legitimate user traffic is forwarded through a router under the control of an adversary). Warnings about attacks exploiting routing vulnerabilities were given as early as 1988 by Perlman [Perlman 1988], and 1989 by Bellovin [Bellovin 1989]; and such

This paper extends the prelimimary work published in NDSS'05 [Wan et al. 2005].

Version: September 20, 2005.

attacks have recently reportedly been carried out by spammers [Bellovin 2004].

Many proposals [Kent et al. 2000; Goodell et al. 2003; White 2003; Aiello et al. 2003] have been made for improving BGP security, and in particular, for verifying if an AS has the right to announce a given IP prefix. There are two main approaches: 1) building centralized routing registries storing information about address space assignments, e.g., Internet Routing Registry (IRR) [IRR 2005]; and 2) building a strict hierarchical public key infrastructure (PKI) in parallel to the existing IP address assignment structure (e.g., S-BGP [Seo et al. 2001; Lynn et al. 2003]). While these two approaches may differ in many ways, e.g., protecting a database itself vs. protecting individual objects in the database, they both typically require a large scale PKI to provide strong security or to meet some operational requirements (e.g., multi-homing). Such a PKI continues to be viewed as impractical by many experts [Atkinson and Floyd 2004].

IRR needs to perform identity authentication to verify if an entity requesting to make changes to the routing database is authorized to do so. Currently in IRR, PGP [Zimmermann 1995] is used for public key authentication. However, this authentication is done using a sender's email address when an object is first created, and thus is vulnerable to email spoofing [Zsako 1999]. As a result, a global PKI or something equivalent, appears to be required to provide stronger guarantees. S-BGP makes use of a hierarchical tree structure for address assignment, rooted at Regional Internet Registries (RIRs). For each consecutive pair of nodes on the address assignment chain, the first node (an organization) on the chain assigns a subset of its own address space to the second one. While an organization obtaining its address space from its Internet Service Providers (ISPs) may not need to appear on an address delegation chain (i.e., need not be issued relevant certificates), it will need those certificates (e.g., a public key certificate and an address assignment certificate) to do multi-homing (i.e., connecting to two independent ISPs). Multi-homing has been considered as a common operational practice which must be supported [Villamizar et al. 1999]. This implies that many organizations not running BGP may also need to be involved in the S-BGP PKI, resulting in a large scale global PKI.

In addition, it appears to be difficult to build a centralized PKI for verifying IP address assignment given the complexity, if not impossibility, of tracing how the existing IP address space is assigned, and tracing all changes of IP address assignments. This is in part due to the large number of prefixes in use and organizations involved, and frequent organization changes (e.g., corporations splitting, merging, bankruptcy, etc.). As pointed by Aiello et al. [Aiello et al. 2003], it is exceptionally difficult to even approximate an IP address delegation graph for the Internet. Therefore, it may well be impossible to build a centralized PKI mirroring such a complex and unknown delegation structure.

Aside from the challenges of requiring a global PKI, many IP addresses were given out before the existing hierarchical address allocation structures were in place. Thus, address assignment chains might not be applicable to them. Fundamentally, all these approaches assume a trusted source of authoritative routing information which allows detecting false prefix announcements. We suggest that such an assumption may not be realistic, or at least it would be very difficult to build an infrastructure to realize it. As noted by Atkinson and Floyd [Atkinson and Floyd 2004] on behalf of the Internet Architecture Board (IAB): "*a recurring challenge with any form of inter-domain routing authentication is that there is no single completely accurate source of truth about which organizations have the authority to advertise which address blocks*".

CONTRIBUTIONS. In this paper, we present a new BGP security proposal – Pretty Secure BGP (psBGP), fleshing out a preliminary overview [Wan et al. 2005]. psBGP includes defenses against falsification of BGP UPDATE messages, and a new approach for verifying the propriety of prefix origin by cross checking information from multiple, ideally independent, sources. Specific psBGP security goals are outlined in §2.3. psBGP is based on the following concepts: 1) there is no universally trusted authority which knows all truth (i.e., all aspects of the factual reality) about prefix assignments on the Internet; 2) some entities may know part of such truth; and 3) corroboration of information from different sources can increase confidence in the assessment of that information. In particular, RIRs are the trusted authority of initial prefix allocations, and some ASes might have partial knowledge of prefix assignments of their direct neighbors.

PSBGP HIGHLIGHTS. The major architectural highlights of psBGP are as follows.

1) psBGP makes use of a *centralized trust model* for AS number authentication. Each AS obtains a public key certificate from one of several trusted certificate authorities (i.e., RIRs), binding an AS number to a public key. We suggest that such a trust model provides best possible authorization of AS number allocation and best possible authenticity of AS public keys. Authentication is usually the first step towards authorization. Without such a guarantee, an attacker may be able to impersonate another AS and thus be able to announce prefixes assigned to the impersonated AS.

2) psBGP makes use of a rating mechanism for flexibility in balancing security and practicality in prefix origin and AS_PATH verification.

3) psBGP makes use of a *decentralized trust model* for verifying the propriety of IP prefix assignment. Each AS periodically issues a digitally signed *Prefix Assertion List (PAL)* consisting of a number of bindings of an AS number and (zero or more) IP prefixes, one such binding for itself and one for each of its neighbors. An assertion made by an AS s_i regarding its own prefixes (*prefix assertion*) lists all prefixes assigned to s_i. An assertion made by s_i for a neighboring AS s_j (*prefix endorsement*) may list all or a subset of the prefixes assigned to s_j. An *AS prefix graph* (see §4.3) is built independently by each AS s_i based on the $PALs$ which s_i has received from other ASes and s_i's ratings of those ASes. An AS prefix graph is then used for evaluating the trustworthiness and preference of a prefix origin by an AS, in conjunction with its local configurable parameters (e.g., its trust in those ASes involved in a prefix assertion, and trust thresholds). In this way, the difficult task of tracing IP address assignments is distributed across ASes on the Internet.

4) psBGP modifies the S-BGP digital signature approach with a rating mechanism and a stepwise approach for verifying AS_PATH integrity. Each AS computes a weight for an AS_PATH based on ratings of the ASes digitally signing the path, and determines whether or not to accept the path based on local parameters. This approach allows an upgrading path to countering increased threats, as recommended in [Bellovin et al. 2005].

Our design is inspired by the referral model widely used in social society for increasing confidence in the truth of a piece of information when a single authoritative source of truth regarding that information is not available. For example, a job applicant is usually required to provide reference letters to allow cross checking the applicant statements on his quality and background. A reference letter should be from an individual who has closely worked with the applicant, e.g., a former supervisor. Similarly in psBGP, each AS should obtain endorsement for its prefix assertions from some ASes which are likely to have, or likely to be reliable sources for, knowledge of its prefix assignment, e.g., a direct neighbor

with which it has a business relationship. An AS choosing to endorse a prefix assertion made by a neighboring AS should carry out some form of due diligence (or other means to increase accountability) to increase confidence in the correctness of that assertion, i.e., to increase its own confidence that the asserted prefix is indeed assigned to the asserting AS. The security assurances of this aspect of psBGP are directly related to the quality of such due diligence, which will impose extra work on BGP operators; this is the price to pay for increased security.

As discussed in what follows, advantages of psBGP include: 1) *simplicity* – it uses a PKI which has a simple structure, a small number of certificate types, and is of manageable size; 2) *effectiveness* – it is designed to successfully defend against selected threats from uncoordinated, misconfigured or malicious BGP speakers; and 3) *incremental deployability* – it can be incrementally deployed with some incremental benefits.

ORGANIZATION. The rest of the paper is organized as follows. §2 defines notation, overviews BGP, discusses BGP threats, and summarizes BGP security goals. psBGP is presented in §3 and §4. Security and operational analysis of psBGP is given in §5 and §6 respectively. A brief review of related work is given in §7. We conclude in §8.

2. BACKGROUND: BGP SECURITY THREATS AND GOALS

After defining notation, we give a brief overview of relevant aspects of BGP, discuss BGP security threats, and summarize five security goals for BGP, for later use in the paper.

NOTATION. A and B denote entities (e.g., an AS or a BGP speaker). X or Y denotes an assertion which is any statement. An assertion may be *proper* or *improper*. We avoid use of the term *true* or *false* since in BGP, it is not always clear that a statement is 100% factual or not. An assertion is proper if it conforms to the rules (e.g., psBGP rules) governing the related entity making that assertion. Table I defines some of the notation used in this paper.

$\mathbb{S}, s_i$	$\mathbb{S}$ is the set of all AS numbers; currently $\mathbb{S} = \{1, \ldots, 2^{16}\}$. $s_i \in \mathbb{S}$ is an AS number.
$\mathbb{P}, f_i$	$\mathbb{P}$ is the set of all IP addresses. $f_i \subseteq \mathbb{P}$ is an IP prefix specifying a range of IP addresses. $f_i = f_j \cup f_k$ if the IP addresses specified by f_i equal those by f_j and f_k combined.
T	an authority with respect to $\mathbb{S}$ and $\mathbb{P}$, e.g., $T \in \{x \mid x$ is an RIR$\}$.
p_k	$p_k = [s_1, s_2, \ldots, s_k]$ is an AS_PATH; s_1 is the first AS inserted onto p_k.
m	$m = (f_1, p_k)$ is a BGP route (a selected part of a BGP UPDATE message).
$N(s_i)$	s_i's neighbors, i.e., the set of ASes with which s_i establishes a BGP session on a regular basis. A given AS s_i may have many BGP speakers, each of which may establish BGP sessions with speakers from many other ASes. $N(s_i)$ is the set of all other such ASes.
$k_A, \overline{k_A}$	A's public and private keys, respectively.
$\{m\}_A$	digital signature on message m generated with A's private key $\overline{k_A}$.
$(k_A, A)_{\overline{k_B}}$	a public key certificate binding k_A to A, signed using $\overline{k_B}$, verifiable using k_B.
$(f_i, s_i)_A$	an assertion made by A that f_i is assigned to s_i.

Table I. Notation

2.1 Selective Overview of BGP

Conceptually, a routing network can be abstracted as a graph, where a vertex is a router and an edge is a network link. If a network consists of a small (e.g., several) or medium (e.g., tens or hundreds) number of routers, a single routing protocol may be capable of exchanging and maintaining routing information in that network. Since there are a large

number of routers (e.g., exceeding hundreds of thousands) on the Internet, any single routing protocol currently available is apparently unable to scale to that size. As a result, a hierarchical routing approach has been used for the Internet. Internet routing protocols can be classified as *intra-domain* (used within an AS) or *inter-domain* (used between ASes).

BGP is an inter-domain routing protocol based on a *distance vector* approach. A BGP speaker establishes a session over TCP with each of its direct neighbors, exchanges routes with them, and builds routing tables based on the routing information received from them. Unlike a simple distance vector routing protocol (e.g., RIP [Hedrick 1988]) where a route has a simple metric (e.g., number of hops), a BGP route is associated with a number of attributes and routes are selected based on local routing policy. One notable route attribute is AS_PATH, which consists of the sequence of ASes traversed by the route that is being propagated. BGP is often considered a *path vector* routing protocol.

ASes on the Internet can be roughly classified into three categories: a *stub-AS* has only one connection to other ASes; a *multihomed-AS* has more than one connection to other ASes, but is not designed to carry traffic for other ASes (e.g., for the purpose of load balance or redundancy); and a *transit-AS* has more than one connection to other ASes, and is designed to carry traffic for others.

While a stub-AS may have only one BGP speaker, a multihomed or a transit-AS often has more. A BGP session between two BGP speakers located within two different ASes is often referred to as external-BGP (eBGP), and a BGP session between two BGP speakers within a common AS is often referred to as internal-BGP (iBGP). An eBGP speaker actively exchanges routing information with an external neighbor by importing and exporting BGP routes. An iBGP speaker only helps propagate routing updates to other BGP speakers within a common AS; it does not make any changes to a routing update.

A BGP session between two different ASes usually implies one of the following four types of business relationship [Gao 2000]: *customer-to-provider*, *provider-to-customer*, *peer-to-peer*, and *sibling-to-sibling*. A customer AS usually pays a provider AS for accessing the rest of the Internet. Two peer ASes usually find it is mutually beneficial to allow each other to have access to their customers. Two sibling ASes are usually owned by a common organization and allow each other to have access to the rest of the Internet.

2.2 BGP Security Threats

BGP faces threats from both BGP speakers and BGP sessions. A misbehaving BGP speaker may be misconfigured (mistakenly or intentionally), compromised (e.g., by exploiting software flaws), or unauthorized (e.g., by exploiting a BGP peer authentication vulnerability). A BGP session may be compromised or unauthorized. We focus on threats against BGP control messages without considering those against data traffic (e.g., malicious packet dropping [Just et al. 2003]). Attacks against BGP control messages include, for example, modification, insertion, deletion, exposure, and replaying of messages. In this paper, we focus on modification and insertion (hereafter *falsification* [Barbir et al. 2004]) of BGP control messages; deletion, exposure and replaying are beyond the scope of this paper, other than the following brief remarks. Deletion appears indistinguishable from legitimate route filtering. Exposure might compromise confidentiality of BGP control messages, which may or may not be a major concern [Barbir et al. 2004]. Replaying is a serious threat, which can be handled by setting an expiration time for each message; however it seems challenging to find an appropriate value for an expiration time.

There are four types of BGP control messages: OPEN, KEEPALIVE, NOTIFICATION,

and UPDATE. The first three are used for establishing and maintaining BGP sessions with peers, and falsification of them will very likely result in session disruption. As mentioned by Hu et al. [Hu et al. 2004], they can be protected by a point-to-point authentication protocol, e.g., IPsec [Kent and Atkinson 1998a]. In psBGP, we concentrate on falsification of BGP UPDATE messages (and hereafter, refrain from capitalizing UPDATE) which carry inter-domain routing information and are used for building up routing tables.

A BGP update message consists of three parts: withdrawn routes, network layer reachability information (NLRI), and path attributes (e.g., AS_PATH, LOCAL_PREF, etc.). As commonly agreed [Hu et al. 2004], a route should only be withdrawn by a party which had previously announced that route. Otherwise, a malicious entity could cause service disruption by withdrawing a route which is actually in service. Digitally signing BGP update messages would allow one to verify if a party has the right to withdraw a route. Further discussion is beyond the scope of the present paper.

NLRI consists of a set of IP prefixes sharing the same characteristics, as described by the path attributes. NLRI is *falsified* if an AS originates a prefix not owned by that AS, or aggregated improperly from other routes. Examples of consequences include denial of service and man-in-the-middle attacks. There are two types of AS_PATH: AS_SEQUENCE and AS_SET. An AS_PATH of type AS_SEQUENCE consists of an ordered list of ASes traversed by the route currently being propagated. An AS_PATH of type AS_SET consists of an unordered list of ASes, sometimes created when multiple routes are aggregated. An AS_PATH is falsified if an AS or any other entity illegally operates on an AS_PATH, e.g., inserting a wrong AS number, deleting or modifying an AS number on the path, etc. Since AS_PATH is used for detecting routing loops and used by route selection processes, falsification of AS_PATH can result in routing loops or selecting routes not selected otherwise. Some other path attributes (e.g., community, Multi_Exit_Disc, etc. [Rekhter and Li 1995]) may also need protection, but many of these are usually only used between two neighbors and not globally transitive. Thus, damage resulting from attacking them is relatively contained. In psBGP, we focus on countering falsification of NLRI and AS_PATH which can result in large scale service disruption.

We assume there are multiple non-colluding misbehaving ASes and BGP speakers in the network, which may have their own legitimate cryptographic keying materials. This non-colluding assumption is also needed by other BGP security proposals (e.g., S-BGP and soBGP), although consequences resulting from collusion might be different.

2.3 BGP Security Goals

We seek to design secure protocol extensions to BGP which can resist the threats as discussed above, i.e., primarily falsification of BGP update messages. As with most other secure communication protocols, BGP security goals must include data origin authentication and data integrity. In addition, verification of the propriety of BGP messages is required to resist falsification attacks. Specifically, the propriety of NLRI and AS_PATH should be verified. All verification will be performed most likely by a BGP speaker online, but possibly by an operator off-line, which is not discussed in the present paper.

We summarize five security goals for BGP (cf. [Kent et al. 2000], also see [Wan et al. 2005; Wan et al. 2005]), for reference later in §3, §4, §5.1 and §7. G1 and G2 relate to data origin authentication, G3 to data integrity, and G4 and G5 to the propriety of BGP control messages. These five security goals address a large number of serious threats against BGP. Thus it is highly desirable for any serious BGP security proposal to achieve them. However,

these alone should not be considered as sufficient for BGP security, since other threats (e.g., replaying) remain (see §2.2).

G1. *(AS Number Authentication)* It must be verifiable that an entity using an AS number s_i as its own is in fact an authorized representative of the AS to which a recognized AS number authority assigned s_i.

G2. *(BGP Speaker Authentication)* It must be verifiable that a BGP speaker, which asserts an association with an AS number s_i, has been authorized by the AS to which s_i was assigned by a recognized AS number authority.

G3. *(Data Integrity)* It must be verifiable that a BGP control message has not been illegally modified en route.

G4. *(AS Path Verification)* It must be verifiable that an AS_PATH ($p_k = [s_1, s_2, \ldots, s_k]$) of a BGP route m being propagated consists of a sequence of ASes traversed by m in the specified order, i.e., m originated from s_1, and has traversed $s_2, \ldots, s_k$ in order.

G5. *(Prefix Origin Authentication)* It must be verifiable that it is proper for an AS to originate an IP prefix. It is *proper* for AS s_1 to originate prefix f_1 if 1) f_1 is indeed assigned to s_1; or 2) s_1 is assigned a set F_1 of prefixes; s_1 has received a set of routes with a set F_2 of prefixes; and f_1 is aggregated from F_1, F_2 or both such that $\forall f_x \subseteq f_1$, $f_x \subseteq F_1 \cup F_2$.[1]

3. PRETTY SECURE BGP (PSBGP)

psBGP makes use of a centralized trust model for authenticating AS numbers and AS public keys. RIRs are the root trusted certificate authorities. In psBGP, each AS s is issued a public key certificate (ASNumCert), signed by one of the RIRs (say T), denoted by $(k_s, s)_{\overline{k_T}}$. Such an AS creates and signs two data structures: a SpeakerCert $(k'_s, s)_{\overline{k_s}}$ binding a different public key k'_s to s; and a *prefix assertion list* (PAL). The latter, pal_s, is an ordered list: the first assertion is for s itself and the rest are endorsements by s for each of s's neighbors ordered by AS number. Figure 1 illustrates the certificate structure used in psBGP. In what follows, we start with a description of a rating mechanism used by each AS in determining its confidence in an AS_PATH or a prefix assertion. We next describe psBGP with respect to the above five security goals: G1-G4 here, and G5 in §4.

3.1 A Rating Mechanism

In psBGP, each AS s_i rates every other AS s_j with a value in $[0, 1]$, denoted by $r_i(s_j)$, representing s_i's confidence or belief in s_j's trustworthiness, i.e., in an assertion made by s_j such as a digitally signed AS_PATH or a prefix assertion or endorsement. $r_i(s_j)$=0 or 1 respectively indicates s_i fully distrusts or trusts s_j. When there is no ambiguity, we omit the subscript on r in $r_i(s_j)$.

While each AS has freedom in determining how to rate other ASes, we suggest the following guidelines: an RIR should be fully trusted (i.e., rated 1); a direct neighbor might be expected, in many cases, to be more trustworthy than a remote AS; and a majority of ASes should be neutrally trusted, e.g., rated 0.5. We next present a method [Wan et al. 2004] for computing the confidence value in a statement which is consistent among a set of assertions made by a group of ASes (a *corroborating* group) based on one's ratings of

[1] If f_1 is not assigned to s_1 and $\exists f_x \subseteq f_1$ such that $f_x \not\subseteq F_1 \cup F_2$, then s_1 *overclaims* IP prefixes, which is a type of falsification.

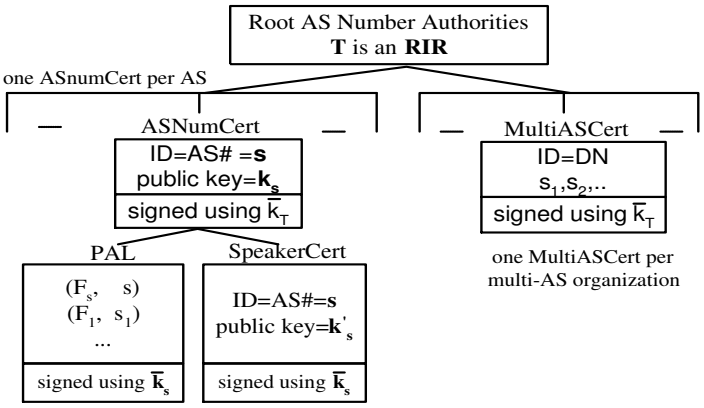

Fig. 1. psBGP Certificate Structure

those ASes. We consider two types of consistency in psBGP: *path-consistency* and *prefix-consistency*. The former is regarding the consistency among a set of digital signatures over an AS_PATH (see Definitions 1 2 in §3.5). The latter is regarding the consistency of a prefix assertion and a prefix endorsement (see Definition 4 in §4.1).

Let $s_1, .., s_n$ be a group of ASes which independently produce a set of consistent assertions $a_{s_1}, .., a_{s_n}$. Let $\lambda_{s_i,..,s_n}$, abbreviated by $\lambda_{[1..n]}$, denote a common subset that can be derived from each of the above n consistent assertions. The precise meaning of $\lambda_{[1..n]}$ depends on the type of consistency in question. In prefix-consistency, if a_{s_1} is a prefix assertion $(f_1, s_1)_{s_1}$, and $a_{s_2}, .., a_{s_n}$ prefix endorsements $(f_1, s_1)_{s_2}, .., (f_1, s_1)_{s_n}$, then $\lambda_{[1..n]}$ represents a prefix assignment of s_1, i.e., s_1 is assigned a prefix f_1. In path-consistency, if $a_{s_1}{=}\{f_1, [s_1, s_2]\}_{s_1}, .., a_{s_n}{=}\{f_1, [s_1, .., s_n, s_{n+1}]\}_{s_n}$ are digital signatures present with a BGP route $m{=}(f_1, p_n{=}[s_1, .., s_n])$, then λ_{s_1,s_2} represents a statement that p_n contains a path segment $[s_1, s_2]$, λ_{s_2,s_3} represents a statement that p_n contains a path segment $[s_2, s_3]$, and so on. We next show how an AS s_i computes a confidence value or a belief in $\lambda_{[1..n]}$, denoted $b(\lambda_{[1..n]})$, based on s_i's ratings of $s_1, .., s_n$ in the corroborating group. By definition, s_i's rating of $s_j, 1{\leq}j{\leq}n$, represents s_i's confidence in the assertion a_j made by s_j or any subset λ_{s_i} derived from a_j, i.e., $b(\lambda_{s_j}){=}b(a_{s_j}) \triangleq r(s_j)$. $b(\lambda_{[1..n]})$ is defined as:

$$b(\lambda_{[1..n]}) = \begin{cases} r(s_1) & \text{if } n{=}1 \\ r(s_2) + \big[1 - r(s_2)\big] \cdot r(s_1) & \text{if } n{=}2 \\ r(s_n) + \big[1 - r(s_n)\big] \cdot b(\lambda_{[1..(n-1)]}) & \text{if } n{\geq}3 \end{cases} \quad (1)$$

Consistent with Dempster-Shafer theory [Dempster 1967; Shafer 1976] of belief reasoning, properties of equation (1) include: i) endorsement from a fully distrusted AS (i.e., rated 0) does not increase one's confidence; ii) endorsement from a fully trusted AS (i.e., rated 1) increases one's confidence to maximum (i.e., 1); and iii) if no AS in the corroborating group is fully distrusted or trusted (i.e., the rating is $0{<}r{<}1$), one's confidence increases but never reaches maximum.

For later cross-reference, Algorithm 1 describes how to increase one's confidence in $\lambda_{[1..(n-1)]}$ when an additional endorsement is obtained, e.g., from s_n. Algorithm 2 describes how to reduce one's confidence in $\lambda_{[1..n]}$ when (without loss of generality) s_n's endorsement is withdrawn.

Algorithm 1 Adding new endorsement from AS s_n

1: **INPUT:** $b(\lambda_{[1..(n-1)]}), r(s_n)$
2: **OUTPUT:** $b(\lambda_{[1..n]})$
3: $t \leftarrow r(s_n) + [1 - r(s_n)] \cdot b(\lambda_{[1..(n-1)]})$
4: return(t)

Algorithm 2 Removing existing endorsement from AS s_n

1: **INPUT:** $b(\lambda_{[1..n]}), r(s_n)$
2: **OUTPUT:** $b(\lambda_{[1..(n-1)]})$
3: $t \leftarrow \frac{b(\lambda_{[1..n]}) - r(s_n)}{1 - r(s_n)}$
4: return(t)

3.2 AS Number Authentication in psBGP (G1)

Following S-BGP [Seo et al. 2001], psBGP makes use of a centralized PKI for AS number authentication, with four root Certificate Authorities (CAs), corresponding to the four existing RIRs. When an organization B applies for an AS number, besides supplying documents currently required (e.g., routing policy, neighboring ASes, etc.), B additionally supplies a public key, and should be required to prove possession of the corresponding private key [Seo et al. 2001; Adams and Lloyd 2003]. When an AS number is granted to B by an RIR, a public key certificate (ASNumCert) is also issued, signed by the issuing RIR, binding the public key supplied by B to the granted AS number. An AS number s is called *certified* if there is a valid ASNumCert $(k_s, s)_{\overline{k_T}}$, binding s to a public key k_s signed by one of the RIRs, T.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
Start of month	16 554	16 708	16 879	17 156	17 350	17 538	17 699	17 884
Removed during month	153	137	155	174	138	179	164	N/A
Added during month	307	308	432	368	326	342	349	N/A

Table II. AS Number Dynamics from January 1 to August 1, 2004

The proposed PKI for authenticating AS numbers is practical for the following reasons. a) The roots of the proposed PKI are the existing trusted authorities of the AS number space, removing a major trust issue which is one of the most difficult parts of a PKI: the root of a PKI must have control over the name space involved in that PKI. Thus, RIRs are the natural and logical AS number certificate authorities. We acknowledge that non-trivial (but feasible) effort might be required for implementing such a PKI. b) The number of ASes on the Internet and its growth rate are relatively manageable (see Table II). Considering there are four RIRs, the overhead of managing ASNumCerts should certainly be manageable, given that larger PKIs are currently commercially operational [Guida et al. 2004].

To verify the authenticity of an ASNumCert, an AS must have the trusted public key (or verifiable certificate) of the signing RIR. These few root trusted public key certificates can be distributed using *out-of-band* mechanisms. ASNumCerts can be distributed with BGP update messages. An ASNumCert should be revoked when the corresponding AS number is no longer used or is reassigned to another organization. Issues of revocation, though

extremely important, are beyond the scope of the present paper; we restrict comment to the observation that revocation is a well-studied, albeit still challenging issue (e.g., see [Adams and Lloyd 2003]). So far, we assume that every AS has the public key certificates of RIRs and can obtain the ASNumCerts of any other ASes if and when necessary.

In discussion related to various proposals for securing BGP, there is much debate in the BGP community on the architecture for authenticating the public keys of ASes, particularly on the pros and cons of using a strict hierarchical trust model vs. a distributed trust model, e.g., a web-of-trust model [Zimmermann 1995]. We make use of a strict hierarchical trust model (with depth of one) for authenticating AS numbers and their public keys to provide a strong guarantee of security. Therefore, it would appear to be difficult for an attacker to spoof an AS in psBGP as long as it cannot obtain the private key corresponding to the public key of an ASNumCert signed by an RIR, or the signing key of an RIR. In contrast, a web-of-trust model does not provide such a guarantee. Other issues that arise with a web-of-trust model include: trust bootstrapping, trust transitivity, and vulnerability to a single misbehaving party [Maurer 1996; Reiter and Stubblebine 1997].

3.3 BGP Speaker Authentication in psBGP (G2)

An AS may have one or more BGP speakers. A BGP speaker must be authorized by an AS to represent that AS to establish a BGP session with a BGP speaker in another AS. In psBGP, an AS with a certified ASNumCert issues an operational public key certificate shared by all BGP speakers within the AS, namely SpeakerCert. A SpeakerCert is signed using the private key of the issuing AS, corresponding to the public key in the AS's ASNumCert (see Figure 1). A SpeakerCert is an assertion made by an AS that a BGP speaker with the corresponding private key is authorized to represent that AS. SpeakerCerts can be distributed with BGP update messages.

We consider three design choices for BGP speaker authentication: a) each BGP speaker has a distinct key pair and is issued a unique public key certificate; b) group signatures (e.g., see [Boneh et al. 2004]) are used, i.e., each BGP speaker has a unique private key but shares a common public key and public key certificate with other speakers in the same AS; or c) all BGP speakers in a given AS share a common public-private key pair. We propose the latter primarily for its operational simplicity. Choice a) provides stronger security in theory but requires more certificates, and discloses BGP speaker identities, which may introduce competitive security concerns [White et al. 2004]. Choice b) again provides stronger security in theory, requires the same number of certificates, and does not disclose BGP speaker identities, but involves a more complex system, which we believe significantly reduces its chances of being commercially accepted and securely deployed.

The private key corresponding to the public key of a SpeakerCert is used for establishing secure connections with neighbors (§3.4), and for signing BGP update messages. Therefore, it would most likely be stored in the communication device associated with a BGP speaker. In contrast, since the private key corresponding to the public key of an ASNumCert is only used for signing a SpeakerCert and a PAL, it need not be stored in a BGP speaker. Thus, compromising a BGP speaker at most discloses the private key of a SpeakerCert, requiring revocation and reissuing of a SpeakerCert, without impact on an ASNumCert. This separation of ASNumCerts from SpeakerCerts provides a more conservative design (from a security viewpoint), and distributes from RIRs to ASes (or their delegated certificate service providers) the workload of certificate revocation and reissuing resulting from BGP speaker compromises. In summary, an ASNumCert must be revoked

if the corresponding AS number is re-assigned or the corresponding key is compromised; a SpeakerCert must be revoked if a BGP speaker in that AS is compromised, or for other reasons (e.g., if the private key is lost).

3.4 Data Integrity in psBGP (G3)

To protect data integrity, BGP sessions between neighboring ASes must be protected. Following S-BGP and soBGP, psBGP uses IPsec Encapsulating Security Payload (ESP) [Kent and Atkinson 1998b] with null encryption for protecting BGP sessions. Since many existing BGP speakers implement TCP MD5 [Heffernan 1998] with manual key configurations for protecting BGP sessions, it must be supported by psBGP as well. In psBGP, automatic key management techniques can be implemented to improve the security of TCP MD5 as each BGP speaker has a public-private key pair (common to all speakers in that AS).

3.5 AS_PATH Verification in psBGP (G4)

Regarding "AS_PATH security", different security solutions of BGP define it differently. In S-BGP, the security of an AS_PATH is interpreted as follows: for every pair of ASes on the path, the first AS authorizes the second to further advertise the prefix associated with this path. In soBGP [White 2003], it is defined as the plausibility of an AS_PATH, i.e., if an AS_PATH factually exists on the AS graph (whether or not that path was actually traversed by an update message in question is irrelevant).

Since AS_PATH is used by the BGP route selection process, greater assurance of the integrity of an AS_PATH increases the probability that routes are selected based on proper information. Without strong guarantees of AS_PATH integrity, an attacker may be able to modify an AS_PATH in a such way that it is still plausible in the AS graph and is also more favored (e.g., with a shorter length) by recipient ASes than the original path. In this way, a recipient AS may be misled to favor a falsified route over correct routes, possibly influencing traffic flow. Thus, in our view, it is not sufficient to verify only the existence/non-existence of an AS_PATH if greater assurance of the integrity of an AS_PATH can be provided at acceptable cost.

We choose the S-BGP approach combined with the rating mechanism described in §3.1 to determine dynamically (at run-time) the number of digital signatures on an AS_PATH to be verified. We first give the definition of *path-consistency*, then present how to calculate a confidence value in an AS_PATH.

DEFINITION 1 (PATH-CONSISTENCY). *Let $m{=}(f_1, p_k{=}[s_1, .., s_k])$ be a BGP route, and $sig_i{=}\{f_1, p_i\}_{s_i}$ be a digital signature generated by a psBGP-enabled BGP speaker in $s_i, 1{\leq}i{\leq}k$, where $\{p_i\}_{s_i}{=}[s'_1, .., s'_{i+1}]$ is the path signed by s_i. $\{p_i\}_{s_i}$ is consistent with p_k if $\{p_i\}_{s_i}$ consists of the first $i{+}1$ ASes on p_k (i.e., $s'_1{=}s_1, .., s'_{i+1}{=}s_{i+1}$) when $1{\leq}i{\leq}k{-}1$, or consists of p_k appended by another AS s_{k+1} when $i{=}k$.*

DEFINITION 2 (SIGNED-PATH CONSISTENCY). *Let $m{=}(f_1, p_k{=}[s_1, .., s_k])$ be a BGP route, and $sig_i{=}\{f_1, p_i\}_{s_i}, sig_j{=}\{f_1, p_j\}_{s_j}$ the digital signatures generated by two psBGP-enabled ASes s_i and s_j, $1{\leq}i, j{\leq}k$, on p_k. $\{p_i\}_{s_i}$ and $\{p_j\}_{s_j}$ are consistent if they both are consistent with p_k.*

Two consistent signed paths $\{p_i\}_{s_i}$ and $\{p_j\}_j$ contain common subset λ_{s_i,s_j}. For example, if $\{p_2\}_{s_2}{=}[s_1, s_2, s_3], \{p_4\}_{s_4}{=}[s_1, s_2, s_3, s_4, s_5]$, λ_{s_2,s_4} could be an assertion that p_k contains the path segment $[s_2, s_3]$ since both s_2 and s_4 assert it in their signed path. As

a result, one may expect the belief in λ_{s_2,s_4} will increase, which may further contribute to the belief in p_k in some way. However, the definition of path confidence in psBGP is more restrictive. In psBGP, the belief in p_k, $b(p_k)$, is defined as the sum of the belief of each assertion that p_k contains a two-AS path segment $[i, i+1]$, $1 \leq i \leq k-1$, divided by the total number of those path segments $k-1$.

DEFINITION 3 (PATH CONFIDENCE). *Let $m=(f_1, p_k=[s_1, .., s_k])$ be a BGP route, and $\lambda_{s_i,s_{i+1}}$ be the assertion that p_k contains a two-AS path segment $[s_i, s_{i+1}]$. The belief in p_k is defined as: $b(p_k) = \frac{1}{k-1} \sum_{i=1}^{i=k-1} b(\lambda_{s_i,s_{i+1}})$.*

The belief in the assertion $\lambda_{s_i,s_{i+1}}$ that p_k contains a two-AS path segment $[s_i, s_{i+1}]$ is obtained exclusively from the signed paths by s_i and s_{i+1} (i.e., $\{p_i\}_{s_i}, \{p_{i+1}\}_{s_{i+1}}$ since two ASes have authority over the path segment between themselves. The signed path by another AS, e.g., s_{i+2}, may also contain $[s_i, s_{i+1}]$, but it does not contribute to the belief in $\lambda_{s_i,s_{i+1}}$ since s_{i+2} apparently does not have authority over $[s_i, s_{i+1}]$ and its signed path may be dependent on the path signed by s_i or s_{i+1}.

If one AS on $[s_i, s_{i+1}]$ is non-psBGP enabled and does not digitally sign its path, the belief in $\lambda s_i, s_{i+1}$ is then solely derived from the signed path of the other AS. If neither of them has signed the path, i.e., $\{p_i\}_{s_i}$ and $\{p_{i+1}\}_{s_{i+1}}$ are null, there is no evidence to believe $\lambda_{s_i,s_{i+1}}$. In this case, $b(\lambda_{s_i,s_{i+1}})$ is set to 0.

In psBGP, a minimum of two digital signatures must be verified if two or more are present on an AS_PATH p_k. The exact number of digital signatures to be verified depends on a verifying AS s_{k+1}'s ratings of the ASes which have signed p_k, and a local configurable confidence threshold $\theta_{k+1} \geq 0$. Verification of p_k starts from the digital signature generated by the last AS s_k on p_k, and moves toward the first AS s_1. Upon a digital signature sig_i verifying successfully, i.e., sig_i is valid and $\{p_i\}_{s_i}$ is consistent with p_k, the belief in the assertion $\lambda_{s_i,s_{i+1}}$ ($1 \leq i \leq k-1$) that p_k contains $[s_i, s_{i+1}]$ is recomputed (using Algorithm 1) and the current belief in p_k is updated (see Definition 3). If $b(p_k)$ is no less then s_{k+1}'s confidence threshold θ_{k+1}, i.e., $b(p_k) \geq \theta_{k+1}$, then p_k is accepted. Otherwise, more digital signatures are verified (see Algorithm 3) until:

a) one digital signature verification fails, in which case p_k is rejected; or

b) $b(p_k) \geq \theta_{k+1}$, in which case p_k is accepted; or

c) all digital signatures present on p_k have been verified successfully, in which case p_k is accepted regardless of $b(p_k)$.

Examining Algorithm 3 (line 5), note that if θ_{k+1} is set to a value higher than 1, then since $0 \leq b(p_k) \leq 1$, $b(p_k)$ will always be less than θ_{k+1}. $i \geq 1$ remains true until all digital signatures are verified. Thus, to always verify all digital signatures present on any received AS_PATH for maximal assurance of path integrity, s_{k+1} can set $\theta_{k+1} > 1$ (e.g., $\theta_{k+1}=1.1$).

If $\theta_{k+1}=0$, $b(p_k) < \theta_{k+1}$ is always false. Once two digital signatures have been verified successfully, $n<2$ remains false. Thus, no additional digital signature will be verified. Such a configuration meets the minimal requirement by psBGP and achieves maximal efficiency. For $0 < \theta_{k+1} \leq 1$, the number of digital signatures on an AS_PATH to be verified depends on s_{k+1}'s rating of each signing AS on the path.

Such configuration flexibility is in line with the recommendation that "a good initial solution is one that can easily be upgraded to handle increased threats" [Bellovin et al. 2005]. For example, an AS with constrained hardware resources (e.g., CPU) can choose

Algorithm 3 AS_PATH Verification (by s_{k+1})

GLOBAL: threshold θ_{k+1}; s_{k+1}'s trust ratings $r(s_1), .., r(s_k)$
INPUT: $k, p_k = [s_1, .., s_k]; sig_1, .., sig_k$
OUTPUT: ACCEPT or REJECT the AS_PATH p_k
$i \leftarrow k; n \leftarrow 0; b \leftarrow 0$ /* b represents $b(p_k)$ */
while $i \geq 1$ and ($b < \theta_{k+1}$ or $n < 2$) **do**
  if $sig_i = \phi$ **then**
    $x \leftarrow 0$ /* s_i has no contribution to belief in λ_{s_{i-1},s_i} or $\lambda_{s_i,s_{i+1}}$ */
  else if sig_i fails verification **then**
    return(REJECT)
  else
    $n \leftarrow n+1; x \leftarrow r(s_i)$
  endif
  if $i = k$ **then**
    $b_2 \leftarrow 0; b_1 \leftarrow x$ /* initial belief in λ_{s_{k-1},s_k} */
  else if $2 \leq i \leq k-1$ **then**
    $b_2 \leftarrow$ Algorithm1(x, b_1) /* final belief in $\lambda_{s_i,s_{i+1}}$ */
    $b_1 \leftarrow x$ /* initial belief in λ_{s_{i-1},s_i} */
  else if $i = 1$ **then**
    $b_2 \leftarrow$ Algorithm1(x, b_1) /* final belief in λ_{s_1,s_2} */
  endif
  $b(p_k) \leftarrow b(p_k) + \frac{b_2}{k-1}$ /* update belief in p_k */
  $i \leftarrow i-1$
return(ACCEPT)

to verify fewer digital signatures on an AS_PATH by setting a lower threshold, while other ASes may choose to verify more or all digital signatures on a signed AS_PATH to achieve a higher assurance of AS_PATH integrity.

We refer to psBGP AS_PATH verification as *stepwise integrity*, which allows confidence ratings on AS_PATH integrity to be formed based on local parameters, and without requiring all ASes on the AS_PATH to digitally sign the path, nor verification of all digital signatures present. In contrast, the S-BGP AS_PATH verification approach provides *full integrity*, but requiring full adoption of S-BGP by all ASes on the path and verification of all digital signatures present.

4. PREFIX ORIGIN AUTHENTICATION IN PSBGP (G5)

We start with descriptions of $PALs$ and MultiASCerts, and then introduce how to build from them an *AS prefix graph*. We then describe how psBGP uses an AS prefix graph to verify the propriety of prefix origin in the two cases per G5 in §2.3.

4.1 Prefix Assertion Lists ($PALs$)

Facing the difficulty of building a centralized infrastructure for tracing changes in IP address assignments (recall §1), psBGP uses a *decentralized* approach for verifying the propriety of a prefix assertion by cross-checking its consistency with endorsements from the neighbors of the asserting AS.

In psBGP, each AS s_i creates and signs an ordered *prefix assertion list* (pal_i), consisting

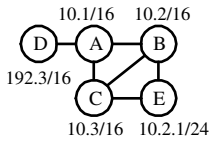

Fig. 2. A small AS graph with IP prefixes and $PALs$ (0 denotes ϕ)

of a number of tuples of the form (*prefixes, AS#*), i.e., $pal_i{=}\{(F_i, s_i), (F_1, s_1), .., (F_n, s_n)\}_{s_i}$, where for the components $(F_j, s_j), 1{\leq}j{\neq}i{\leq}n,\ s_j{\in}N(s_i)$ and $s_j{<}s_{j+1}$. The first tuple (F_i, s_i) is an assertion by s_i of its own assigned prefixes F_i (referred to as *prefix assertions*); the rest are ordered by AS number, and are assertions by s_i of prefixes assigned to each of s_i's neighbors (referred to as *prefix endorsements*). If s_i chooses not to endorse any prefix for a neighbor s_j or has no information of s_j's prefix assignments, s_i simply declares null in its prefix endorsement for s_j. Thus, $(F_j, s_j)_{s_i}$ $(F_j{=}\phi)$ simply asserts that s_j is a direct neighbor of s_i (see Figure 2). If s_i is not willing to disclose that s_j is a direct neighbor, s_i can leave out from pal_i the prefix endorsement for s_j.

DEFINITION 4 PREFIX-CONSISTENCY. *Let $(f_i, s_i)_{s_i}$ be a prefix assertion by s_i and $(f'_i, s'_i)_{s_j}$ a prefix endorsement by s_j. $(f'_i, s'_i)_{s_j}$ is consistent with $(f_i, s_i)_{s_i}$, denoted by $(f'_i, s'_i)_{s_j} \doteq (f_i, s_i)_{s_i}$, if they are regarding the prefix assignment of the same AS, i.e., $s'_i{=}s_i$, and f'_i is equal to or a superset of f_i, i.e., $f'_i{\supseteq}f_i$.*

Inferred from Definition 4, $(f'_i, s'_i)_{s_j}$ is not consistent with $(f_i, s_i)_{s_i}$, if 1) they are regarding the prefix assignment of different ASes; 2) they have null mutual intersection, i.e., $f'_i{\cap}f_i{=}\phi$; or 3) f'_i is a proper subset of f_i, i.e., $f'_i{\subset}f_i$. In case 3, while f'_i and f_i do share a common subset which is f'_i, they are not considered consistent in psBGP for the sake of simplicity of AS prefix graph maintenance. In psBGP, prefix consistency is checked between a prefix assertion and an endorsement, but not between two prefix endorsements.

While an AS is free to decide for which neighbors it provides prefix endorsements and from which to solicit prefix endorsements for itself, we recommend that a provider AS endorse prefixes for a customer AS, possibly becoming a part of an existing service agreement which includes not only physical network connectivity but now also prefix endorsements. Two neighboring ASes with a peer relationship have freedom to decide how one will endorse prefix assertions made by the other. Prefix endorsements between two peering ASes might be *asymmetric*; in the extreme case, AS s_i may endorse all prefixes assigned to a peering AS s_j, while s_j endorses no prefix assigned to s_i. It is important to allow such flexibility. In the core of the Internet, one AS may peer with many others, some of which may be assigned a large number of prefixes. It would be unrealistic to expect an AS to have full knowledge of all prefixes assigned to such a peer. However, an AS might be able to establish a certain level of confidence in a subset of the prefixes assigned to some of its neighbors. Thus, an AS can distribute such positive (albeit partial) evidence to facilitate other ASes to make a better decision in prefix origin authentication. It is an AS's own responsibility and in its own interest to ensure that its assigned prefixes are endorsed by some of its neighbors or by an RIR.

As a new requirement in psBGP, each AS is responsible for carrying out some level of due diligence off-line: for the safety of that AS and of the whole Internet, to increase its confidence that the prefixes it endorses for a direct neighbor are indeed assigned to that AS. We suggest the effort required for this is both justifiable and practical, since two

neighboring ASes usually have a business relationship (e.g., a traffic agreement) with each other, allowing some level of off-line direct interactions and the establishment of some level of trust. For example, s_i may ask a neighboring AS s_j to show the proof that a prefix f_j is in fact assigned to s_j, or may ask a senior official of the neighboring AS organization to provide a formal letter asserting the organization's prefix claim. Publicly available information about IP address allocation and delegation may also be helpful.

A PAL may be distributed along with BGP update messages in newly defined path attributes [Kent 2003], which are optional and transitive. A non-psBGP enabled BGP speaker which does not understand these newly defined attributes need not process them but must propagate them. Thus, $PALs$ travel through non-psBGP enabled BGP speakers to reach psBGP-enabled ones. Each psBGP-enabled BGP speaker can then construct and update its AS prefix graph from received $PALs$ (see §4.3).

4.2 Multiple-AS Certificate (MultiASCerts)

Ideally, two $PALs$ issued by two neighboring ASes are based on independent data sources, and consequently, with high probability (in the absence of collusion), a prefix erroneously asserted by one AS will not be endorsed by any of its neighbors. However, there are some organizations owning multiple ASes, and it is a common practice for a multi-AS organization to use a single centralized database for generating router configurations for all of its owned ASes. Thus, it is possible that $PALs$ issued by two neighboring ASes owned by a common organization would also be created from a single centralized database. If a prefix is erroneously entered into such a database, it might end up with a pair of erroneous yet consistent prefix assertion and endorsement, introducing a single point of failure. We recommend that "best practice" in psBGP requires that an AS obtain prefix endorsement from another AS owned by a different organization. As a recommended BGP local policy, an AS should ignore a prefix endorsement by s_j for s_i if both s_i and s_j are known to be owned by a common organization.

To facilitate the distribution of the knowledge of AS ownership by a multi-AS organization, psBGP makes use of a new certificate, namely MultiASCert (recall Figure 1), which binds a list of ASes owned by a common organization to the name of that organization, and is signed by an RIR. Prefix endorsements by s_j for s_i should be ignored if s_i and s_j appear on a MultiASCert. In this way, human errors by a multi-AS organization regarding a prefix that is assigned to another psBGP-enabled AS and endorsed by an independent neighboring AS will not result in service disruption of that prefix in psBGP (see §4.4.1).

4.3 AS Prefix Graph

We introduce as a new concept the *AS prefix graph*, which contains information about *AS connectivity*, *AS prefix assignments* (or prefix-AS bindings), and *ratings* of AS prefix assignments. An AS prefix graph, constructed by each AS s_c, is an attributed graph $G_c{=}(V, E, H)$, where $V{=}\{s_i\}$ is a set of AS numbers, $E{=}\{e_{ij}\}$ is a set of edges (BGP sessions) with e_{ij} connecting s_i to s_j, and $H{:}\, V{\rightarrow}W$ is a function mapping each AS s_i to a set of three-dimensional variables, which specifies the IP prefixes asserted by s_i, and supporting evidence; we call $H(s_i)$ the *APAS set* (associated prefixes and support) for s_i. More precisely, $H(s_i){=}\{(f_x, b_x, C_x)\}$, where $f_x{\subseteq}\mathbb{P}$ is an IP prefix, $b_x{\in}[0, 1]$ represents s_c's confidence that f_x is assigned to s_i, and C_x is a list of ASes asserting and endorsing the prefix assignment (f_x, s_i). We next present how each psBGP-enabled AS constructs and updates its own AS prefix graph based on the $PALs$ and MultiASCerts it has received.

4.3.1 *AS Prefix Graph Construction.* An AS prefix graph is initialized to null before the BGP speaker receives any PAL (e.g., when it first connects to the Internet). All BGP speakers within an AS build their own AS prefix graph independently. An AS s_c builds its AS prefix graph $G_c{=}(V, E, H)$ from the first pal_i received from each s_i on the Internet by performing the following tasks: a) adding s_i and all of its declared neighbors to V; b) adding to E an edge from s_i to each of its declared neighbors; c) updating $H(s_i)$ for each of the prefixes asserted by s_i; d) updating $H(s_j)$ for each of the prefixes asserted by $s_j{\in}N(s_i)$ and endorsed by s_i. See Algorithm 4 for the details and §4.3.3 for an example.

Algorithm 4 AS Prefix Graph Construction (for AS s_c)

GLOBAL: $G_c{=}(V, E, H)$; existing $PALs$; $\{r_c(s_i)|s_i$ is an AS on the Internet$\}$
INPUT: pal_i
OUTPUT: updated AS prefix graph G_c
/* $F_i, N(s_i)$ are prefixes and neighbors asserted by s_i for itself in pal_i respectively */
$V \leftarrow V{\cup}s_i$; $H(s_i) \leftarrow \phi$
for each $f_x \in F_i$ **do**
  $(f_x, b_x, C_x) \leftarrow (f_x, r(s_i), \{s_i\})$
  for each $s_j \in N(s_i)$ **do**
    $V \leftarrow V{\cup}s_j$; $E \leftarrow E{\cup}e_{ij}$
    for each prefix endorsement $(f, s)_{s_j}$ in pal_j **do**
      /* recall Definition 4 */
      if $(f, s)_{s_j} \doteq (f_x, s_i)_{s_i}$ and s_i, s_j are not in a common MultiASCert **then**
        $b_x \leftarrow$ Algorithm1$\big(b_x, r(s_j)\big)$; $C_x \leftarrow C_x{\cup}s_j$
  $H(s_i) \leftarrow H(s_i){\cup}(f_x, b_x, C_x)$;
for each $s_j \in N(s_i)$ **do**
  retrieve APAS set $H(s_j) = \{(f_y, b_y, C_y)\}$
  for each $(f_y, b_y, C_y) \in H(s_j)$ **do**
    for each prefix endorsement $(f, s)_{s_i}$ in pal_i **do**
      if $(f, s)_{s_i} \doteq (f_y, s_j)_{s_j}$ and s_i, s_j are not in a common MultiASCert **then**
        $b_y \leftarrow$ Algorithm1$\big(b_y, r(s_i)\big)$; $C_y \leftarrow C_y{\cup}s_i$
    $H(s_j) \leftarrow H(s_j){\cup}(f_y, b_y, C_y)$
return

4.3.2 *AS Prefix Graph Update.* Here we describe how to update an AS prefix graph from a newly received pal_i' which replaces an existing pal_i that has been previously used to construct or update an AS prefix graph. The prefix-AS bindings in pal_i and pal_i' can be divided into three categories: *removed*, *unchanged*, and *added*. A removed prefix-AS binding appears in pal_i but not in pal_i'; an unchanged one appears in both; and a newly added one appears in pal_i' but not in pal_i. Updating an AS prefix graph is performed in two phases (see Algorithm 5 for details) as follows:

(1) *Removing prefix-AS bindings.* If a removed prefix-AS binding is an assertion, $(f_x, s_i)_{s_i}$, made by s_i for itself, it is simply removed from the graph. If it is an endorsement, $(f_y, s_j)_{s_i}$, by s_i for $s_j{\in}N(s_i)$, the confidence in s_j's assertion of f_y must be updated (using Algorithm 2).

(2) *Adding prefix-AS bindings*. If an added prefix-AS binding is an assertion, $(f_x, s_i)_{s_i}$, made by s_i for itself, a confidence value must be computed for $(f_x, s_i)_{s_i}$ (using Algorithm 1). If it is a prefix endorsement, $(f_y, s_j)_{s_i}$, and $(f_y, s_j)_{s_j}$ exists in the graph, the confidence in $(f_y, s_j)_{s_j}$ must be updated (using Algorithm 1).

Algorithm 5 AS Prefix Graph Update (for AS s_c)

GLOBAL: G_c=(V, E, H); existing $PALs$; $\{r_c(s_i)|s_i$ is an AS on the Internet$\}$
INPUT: pal_i'
OUTPUT: updated AS prefix graph G_c
/* $N(s_i)'$ is the set of neighbors asserted by s_i for itself in pal_i' */
/* Removing prefix-AS bindings */
for each prefix assertion $(f_x, s_i)_{s_i}$ in pal_i that is not in pal_i' **do**
  retrieve the APAS set $H(s_i) = \{(f_x, b_x, C_x)\}$
  $H(s_i) \leftarrow H(s_i) - (f_x, b_x, C_x)$ /* set subtraction */
for each prefix endorsement $(f_y, s_j)_{s_i}$ in pal_i that is not in pal_i' **do**
  retrieve the APAS set $H(s_j) = \{(f_y, b_y, C_y)\}$
  if $H(s_j) \neq \phi$ and $s_i \in C_y$ **then**
    $b_y \leftarrow$ Algorithm2$\big(b_y, r(s_i)\big)$; $C_y \leftarrow C_y - s_i$
for each s_j in $N(s_i)$ that is not in $N(s_i)'$ **do**
  $E \leftarrow E - e_{ij}$
  if s_j has no neighbor or prefix assignment in G_c **then**
    $V \leftarrow V - s_j$
/* Adding prefix-AS bindings */
for each s_j in $N(s_i)'$ that is not in $N(s_i)$ **do**
  $V \leftarrow V \cup s_j$; $E \leftarrow E \cup e_{ij}$
for each prefix assertion $(f_x, s_i)_{s_i}$ in pal_i' that is not in pal_i **do**
  $(f_x, b_x, C_x) \leftarrow (f_x, r(s_i), \{s_i\})$
  for each $s_j \in N(s_i)'$ **do**
    for each prefix endorsement $(f, s)_{s_j}$ in pal_j **do**
      if $(f, s)_{s_j} \doteq (f_x, s_i)_{s_i}$ and s_i, s_j are not in a common MultiASCert **then**
        $b_x \leftarrow$ Algorithm1$\big(b_x, r(s_j)\big)$; $C_x \leftarrow C_x \cup s_j$
  $H(s_i) \leftarrow H(s_i) \cup (f_x, b_x, C_x)$
for each $s_j \in N(s_i)'$ **do**
  for each prefix endorsement $(f, s_j)_{s_i} \in pal_i'$ that is not in pal_i **do**
    retrieve APAS set $H(s_j) = \{(f_y, b_y, C_y)\}$
    for each $(f_y, b_y, C_y) \in H(s_j)$ **do**
      if $(f, s_j)_{s_i} \doteq (f_y, s_j)_{s_j}$ and s_i, s_j are not in a common MultiASCert **then**
        $b_y \leftarrow$ Algorithm1$\big(b_y, r(s_i)\big)$; $C_y \leftarrow C_y \cup s_i$
return

4.3.3 *Example 1*. Figure 3 illustrates Algorithm 4 for an AS D. Assume D fully trusts its service provider A (i.e., $r(A)$=1), and partially trusts the other ASes ($r(B)$=$r(E)$=0.5, $r(C)$=0.8). The AS prefix graph is constructed based on the following $PALs$ received by D in order (here we focus on the construction of the APAS set):

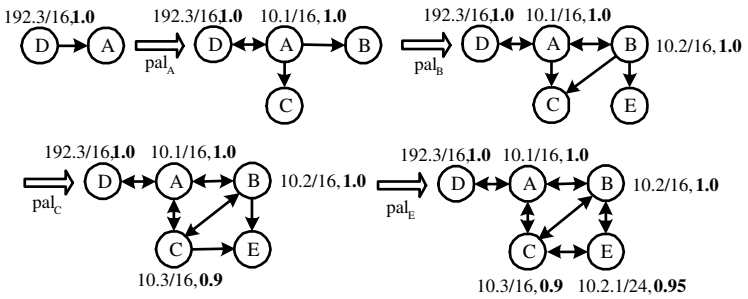

Fig. 3. Construction of an AS Prefix Graph by AS D (see Example 1)

$$pal_D{=}\{(192.3/16, D), (\phi, A)\}_D,$$
$$pal_A{=}\{(10.1/16, A), (10.2/16, B), (\phi, C), (192.3/16, D)\}_A,$$
$$pal_B{=}\{(10.2/16, B), (\phi, A), (10.3/16, C), (10.2.1/24, E)\}_B,$$
$$pal_C{=}\{(10.3/16, C), (10.1/16, A), (\phi, B), (10.2.1/24, E)\}_C,$$
$$pal_E{=}\{(10.2.1/24, E), (\phi, B), (\phi, C)\}_E.$$

1) D starts from pal_D issued by itself, and updates the graph as: $V{=}\{D, A\}$; $E{=}\{e_{DA}\}$; and $H(D){=}\{(192.3/16, 1.0, \{D\})\}$. After receiving pal_A, D initializes $H(A)$ to $\{(10.1/16, 1.0, \{A\})\}$ (Algorithm 4 (line 7)). Since A endorses D's prefix assertion, $H(D)$ is updated to $\{(192.3/16, 1.0, \{D, A\})\}$. While A also endorses B's prefix assertion, no action is taken at this time since D has not received pal_B.
2) After receiving pal_B, D initializes $H(B){=}\{(10.2/16, 0.5, \{B\})\}$. Since A endorses $(10.2/16, B)$, Algorithm1$(0.5, 1.0)$ is called to update D's confidence in $(10.2/16, B)$, and $H(B)$ is updated to $\{(10.2/16, 1.0, \{B, A\})\}$.
3) After receiving pal_C, D initializes $H(C){=}\{(10.3/16, 0.8, \{C\})\}$. Since B endorses $(10.3/16, C)$, Algorithm1$(0.8, 0.5)$ is called to update D's confidence in $(10.3/16, C)$ to 0.9, and $H(C)$ is updated to $\{(10.3/16, 0.9, \{C, B\})\}$. Since C endorses A's prefix assertion, Algorithm1$(1.0, 0.8)$ is called to update D's confidence in $(10.1/16, A)$, which does not change since it already has maximal value 1.0 (see above). $H(A)$ is updated to $\{(10.1/16, 1.0, \{A, C\})\}$.
4) After receiving pal_E, D initializes $H(E){=}\{(10.2.1/24, 0.5, \{E\})\}$. Since B endorses $(10.2.1/24, E)$, Algorithm1$(0.5, 0.5)$ is called to update D's confidence in $(10.2.1/24, E)$ to 0.75. Since C also endorses $(10.2.1/24, E)$, Algorithm1$(0.75, 0.8)$ is called to further update D's confidence in $(10.2.1/24, E)$ to 0.95. As a result, $H(E)$ is updated to $\{(10.2.1/24, 0.95, \{E, B, C\})\}$.

4.4 Prefix Origin Authentication

Here we describe how to perform prefix origin authentication using an AS prefix graph.

4.4.1 *Verification of Prefix Assignment.* Two configurable thresholds, denoted by α_i (*sufficient confidence*) and β_i (*sufficient claimants*), are used by each psBGP-enabled AS s_i for verifying the propriety of prefix assignments. α_i is a threshold defining a sufficient confidence level by s_i in a prefix-AS binding before it can be considered proper. β_i defines a sufficient number of ASes which assert and endorse a prefix-AS binding before the binding can be considered proper by s_i. In other words, a prefix-AS binding (f_j, s_j) is

verified as proper by s_i if s_i's confidence in (f_j, s_j) is at least α_i, or (f_j, s_j) is asserted by s_j and endorsed by at least $\beta_i - 1$ other ASes. More specifically, a non-aggregated route $(f, [s_j, ..])$ originated by a psBGP-enabled AS s_j is verified by another psBGP-enabled AS s_i as *proper* if a) there exists $(f_x, b_x, C_x) \in H(s_j)$; b) $b_x \geq \alpha_i$ or $|C_x| \geq \beta_i$; and c) $f \subseteq f_x$. Algorithm 6 specifies this explicitly.

Algorithm 6 Verification of Prefix Assignment (by an AS s_i)

```
1: GLOBAL: G_i = (V, E, H); α_i; β_i
2: INPUT: The BGP route m = (f_j, p = [s_j, ..])
3: OUTPUT: ACCEPT or REJECT s_j's origin of f_j
4: retrieve the APAS set H(s_j) = {(f_x, b_x, C_x)} from G_i
5: for each (f_x, b_x, C_x) ∈ H(s_j) do
6:    if (b_x ≥ α_i or |C_x| ≥ β_i) and f_j ⊆ f_x then
7:       return(ACCEPT)
8: return(REJECT)
```

α_i and β_i are independent and in conjunction provide extensive flexibility. $\alpha_i = 1$ allows s_i to immediately accept a prefix assertion by a fully trusted AS (i.e., without any neighbor endorsement), while prefix assertions made by partially trusted ASes require endorsements from a sufficient number of neighbors. α_i and β_i can also be configured such that only one or neither takes effect. For example, $\alpha_i > 1$ and $\beta_i \geq 1$ allows β_i to always take precedence since the maximum confidence in a prefix assertion is 1. $0 < \alpha_i \leq 1$ and $\beta_i = \infty$ has the opposite effect. $\alpha_i = 0$ and $\beta_i = 0$ emulate the existing non-secured BGP behavior (i.e., any prefix originated by any AS is considered as proper).

During the early stages of psBGP deployment, when only a small number of ASes have deployed psBGP, we recommend $\beta_i = 1$ for each psBGP-enabled AS s_i. In other words, a psBGP-enabled AS s_i allows another psBGP-enabled AS s_j to originate a prefix f_j if f_j is asserted in pal_j even it is not endorsed by any neighbor. This reflects the reality that early psBGP adopters might not have any psBGP-enabled neighbors, and it offers some level of assurance (albeit limited). For example, a compromised BGP speaker within a psBGP-enabled AS s_j cannot be used to hijack prefixes assigned to other ASes unless keying material required for issuing pal_j is also compromised. In addition, the existence of a public statement about an assertion provides some assurance, in that this might carry some weight in legal dispute or affect business reputation. See §6.1.2 for more discussion on incremental benefits and §5.2.3 on limitations of psBGP.

After a majority of ASes have deployed psBGP, we recommend $\beta_i = 2$, i.e., a psBGP-enabled AS s_i allows another psBGP-enabled AS s_j to originate a prefix f_j only if f_j is asserted in pal_j and is endorsed by one of s_j's neighbors. $\beta_i = 2$ is resilient to some errors resulting from a single AS. For example, if s_j mistakenly asserts a prefix f in pal_j and announces f via BGP, this would not result in service disruption of the legitimate owner of f as long as s_j's assertion of f is not endorsed by any neighbor. However, $\beta_i = 2$ remains vulnerable to two-party collusion. More generally, $\beta_i = k \geq 2$ resists collusion by $k - 1$ parties. Larger β_i renders a stronger assurance in the propriety of a prefix assignment, but trades off performance and results in higher maintenance overhead (see §6.3.4).

4.4.2 *Verification of Prefix Aggregation.* Suppose AS s_1 is assigned a set of prefixes F_1. When receiving a set of routes with a set of prefixes F_2, the BGP specification [Rekhter and Li 1995] allows s_1 to aggregate F_2 into a single prefix f_g to reduce routing information to be stored and transmitted. We call f_g an *aggregated prefix*. s_1 can aggregate F_2 into f_g if one of the following conditions holds: 1) $\forall f_i \subseteq f_g$, $f_i \subseteq F_1$; or 2) $\forall f_i \subseteq f_g$, $f_i \subseteq F_1 \cup F_2$.

In case 1), s_1 must be assigned a set of prefixes F_1, which is a superset of the aggregated prefix f_g. Most likely, f_g is one of the prefixes assigned to s_1, i.e., $f_g \in F_1$. This type of aggregation is sometimes referred to as prefix *re-origination*. From a routing perspective, prefix re-origination does not have any effect since traffic destined to a more specific prefix will be forwarded to the re-originating AS and then forwarded to the ultimate destination from there. From a policy enforcement perspective, prefix re-origination does have an effect since the AS_PATH of an aggregated route is different from any of the AS_PATHs of the routes to be aggregated. Since AS_PATH is used by the route selection process, changing AS_PATH has an impact on route selections. From a security perspective, prefix re-origination is no different than normal prefix origination since the aggregated prefix is either the same as, or a subset of, the prefix assigned by the aggregating AS. Therefore, f_g can be verified using the mechanism in §4.4.1.

In case 2), s_1 is not assigned the whole address space of the aggregated prefix f_g. Therefore, f_g cannot be verified in the same way as for prefix re-origination. To facilitate verification of the propriety of route aggregation by a receiving AS, psBGP imposes a new requirement: the routes to be aggregated must be supplied by the aggregating AS along with the aggregated route. This approach is essentially similar to that taken by S-BGP. Transmission of routes to be aggregated incurs additional network overhead, which is something BGP tries to reduce. However, we view such additional overhead to be relatively insignificant given that modern communication networks generally have high bandwidth and BGP control messages account for only a small fraction of subscriber traffic. The main purpose of route aggregation is to reduce the size of routing tables, i.e., reducing storage requirements; note that this is preserved by psBGP.

4.5 Route Selection Algorithm

In standard BGP, when a BGP speaker receives two valid routes with the same destination prefix, a route selection process is invoked to determine which is preferable. In what follows, a prefix-AS binding of a route means the binding of the prefix and the AS that originates that route. psBGP adds two new rules: one gives preference to a route whose prefix-AS binding has more neighbor endorsements, and the other to a route whose prefix-AS binding is rated higher. These two new rules are added into the fourth and fifth places in BGP route selection algorithm [Rekhter and Li 1995] to preserve existing traffic engineering practices which usually employ $local\_pref$, as_path and med ($mult\_exit\_disc$). Note that the higher-numbered rule is followed if the lower-numbered rules result in a tie.

1) Select the route with a higher degree of preference, i.e., a higher $local\_pref$ value.
2) Select the route with a shorter as_path.
3) Select the route with a lower med value if they have the same $next\_hop$.
4) *Select the route whose prefix-AS binding is endorsed by more neighbors.*
5) *Select the route whose prefix-AS binding is rated higher.*
6) Select the route with a lower intra-domain routing cost to the $next\_hop$.

Ongoing work [Retana and White 2002] suggests to allow customer-defined rules to be inserted anywhere in the standard BGP route selection algorithm. If this is implemented in psBGP, customers with high security requirement can choose to move psBGP-related rules up to an appropriate decision point, e.g., as rules 1 and 2.

5. SECURITY ANALYSIS OF PSBGP

We first analyze psBGP against the listed security goals from §2. We then discuss how psBGP counters selected BGP threats.

5.1 Meeting Specified Security Goals

The analysis below clarifies how the proposed psBGP mechanisms meet the specified goals, and by what line of reasoning and assumptions. While we believe that mathematical "proofs" of security may often be based on flawed assumptions or models (e.g., see [Koblitz and Menezes 2004]) that fail to guarantee "security" in any real-world sense, they are nevertheless very useful, e.g., for finding security flaws, for precisely capturing protocol goals, and for reducing ambiguity, all of which increase confidence. We thus provide outlines of such formalized reasoning, as a complement to alternative methods of increasing confidence.

PROPOSITION 1. *psBGP provides AS number authentication (G1).*

Proof Outline: For an AS number s to be certified, psBGP requires an ASNumCert $(k_s, s)_{\overline{k_T}}$. Since T (i.e., an RIR) controls s, and is the trusted guardian of AS numbers (by assumption), any assertion made by T about s is proper. Thus $(k_s, s)_{\overline{k_T}}$ is proper. In other words, s is an AS number certified by T, and k_s is a public key associated with s certified by T. More formally,[2] (T *controls* s) $\wedge$ $(k_s, s)_{\overline{k_T}} \Rightarrow (k_s, s)$ is a proper binding.

PROPOSITION 2. *psBGP provides BGP speaker authentication (G2).*

Proof Outline: For a BGP speaker g to be accepted as an authorized representative of an AS s, psBGP requires an ASNumCert $(k_s, s)_{\overline{k_T}}$, a SpeakerCert $(k'_s, s)_{\overline{k_s}}$, and evidence that g possesses $\overline{k'_s}$. By Proposition 1, $(k_s, s)_{\overline{k_T}}$ establishes that s is an AS number certified by T and k_s is a public key associated with s certified by T. Similarly, $(k'_s, s)_{\overline{k_s}}$ establishes that k'_s is a public key associated with s certified by s. Evidence that g possesses $\overline{k'_s}$ (i.e., an appropriate digital signature generated by g using $\overline{k'_s}$) establishes that g is authorized by s to represent s. Thus, the Proposition is established. More formally, (T *controls* s) $\wedge$ $(k_s, s)_{\overline{k_T}} \Rightarrow (k_s, s)$ is a proper binding; (k_s, s) is proper $\wedge$ $(k'_s, s)_{k_s} \Rightarrow (k'_s, s)$ is proper binding; (k'_s, s) is proper $\wedge$ g possesses $\overline{k'_s} \Rightarrow g$ is authorized by s.

PROPOSITION 3. *psBGP provides data integrity (G3).*

Proof Outline: psBGP uses the IPsec Encapsulating Security Payload (ESP) [Kent and Atkinson 1998b] with null encryption for protecting BGP sessions, and relies upon IPsec ESP for data integrity. Thus this provides data integrity in practice, to the extent that one can rely on practical implementations of IPsec ESP.

PROPOSITION 4. *psBGP provides assurance of AS_PATH authentication (G4).*

[2]Here we adapt BAN-like notation, modified for our purpose (cf. [Burrows et al. 1989; Gaarder and Snekkenes 1991; Gligor et al. 1991]).

Proof Outline: Let m_k=(f_1, p_k) be a BGP route, where p_k=$[s_1, .., s_k]$, and m_k is originated or forwarded by a BGP speaker in s_k. For simplicity, we refer to an AS instead of a BGP speaker within that AS. In psBGP, the integrity of p_k implies that m_k has traversed the exact sequence of $s_1, .., s_k$. We next use induction on path length to show that psBGP provides AS_PATH integrity when all ASes on an AS_PATH are psBGP-enabled and the verifying AS chooses to verify all digital signatures on the path, followed by discussion of other cases.

(1) If k=1, psBGP requires that for s_2 to accept m_1, s_2 must receive a valid digital signature $sig_1 = \{f_1, [s_1, s_2]\}_{s_1}$, which serves as a signed assertion that s_1 originated m_1 (and advertised it to s_2).

(2) Assume when k=n≥2, there exist digital signatures $sig_1, .., sig_n$ which assert that m_n indeed traversed the exact sequence of $s_1, .., s_n$. When k=n+1, we need to show that m_{n+1} has traversed from s_n to s_{n+1} and exited s_{n+1}. $sig_n = \{f_1, [s_1, .., s_n, s_{n+1}]\}_{s_n}$ asserts that s_n forwards m_n to s_{n+1}. psBGP requires that s_{n+1} digitally signs m_{n+1} by generating a digital signature $sig_{n+1} = \{f_1, [s_1, ..., s_{n+1}, s_{n+2}]\}_{s_{n+1}}$, which serves as the evidence that m_{n+1} is advertised by s_{n+1} to another AS s_{n+2}. In summary, sig_n asserts that m_n traversed from s_n to s_{n+1}, and sig_{n+1} asserts that m_n is transformed by s_{n+1} to m_{n+1} which traversed through s_{n+1} to another AS. Thus, the above three steps establish Proposition 4 when all ASes on an AS_PATH are psBGP-enabled and the verifying AS verified all digital signatures on the path.

Partial AS_PATH integrity. If an AS chooses not to always verify all digital signatures on the path (i.e., setting θ<1, or some digital signatures are missing; see Algorithm 3 and §3.5), full integrity of the path is not guaranteed. For example, let p_k=$[s_1, .., s_j, .., s_k]$. If an AS only verifies the digital signatures generated by ASes from s_j to s_k, only the integrity of that the path segment is protected. The path from s_1 to s_{j-1} can be falsified if all ASes from s_j to s_k are in collusion. As another example, consider the route m=$(f, [s_1, s_2, s_3, s_4])$ with only s_2 psBGP-enabled. The digital signature generated by a well-behaved s_2, $\{f, [s_1, s_2, s_3]\}_{s_2}$, covers the path $[s_1, s_2, s_3]$. In other words, a malicious AS cannot compromise the integrity of $[s_1, s_2, s_3]$, but it can insert any non-psBGP enabled AS after s_3 or modify s_4 to another non-psBGP enabled AS. In addition, $[s_1, s_2, s_3]$ can be removed or replaced as a whole with other non-psBGP enabled ASes.

We next establish Proposition 5. As discussed in §3.1, psBGP uses a rating mechanism to provide the flexibility to allow an AS to fully trust an AS or an RIR, thus accepting their prefix assertions without requiring additional endorsements. We recommend that no AS should be fully trusted unless there is strong reason to do so. In the rest of our analysis, we assume that a verifying AS s_i does not immediately trust any other AS s_j. In other words, s_i rates every other AS s_j with a value lower than its confidence threshold, i.e., $r_i(s_j)$<α_i. Before presenting Proposition 5, we establish two Lemmas.

LEMMA 1. *Assuming that no two ASes are in collusion (A1),*[3] *then psBGP with threshold β=2 provides reasonable*[4] *assurance of prefix assignment verification, i.e., a prefix assignment that is verified as proper is, with reasonable assurance, proper.*

[3] See §5.2.3 for discussion of examples where this collusion assumption (A1) may not hold.

[4] By reasonable, we mean to emphasize that our claim is relative to our threat model and assumptions (e.g., see §5.2.3); we cannot claim absolute security (which we do not believe exists in the real world.

Proof Outline: Consider the BGP route $m{=}(f_x, [s_i, ..])$. For f_x to be verified as assigned to s_i, psBGP requires that for some f_i:

(**R1**) prefix assertion $(f_i, s_i)_{s_i}$ exists; (**R2**) $(f_i', s_i)_{s_j} \doteq (f_i, s_i)_{s_i}$ exists for $s_j \in N(s_i)$;
(**R3**) s_i, s_j do not appear in a common MultiASCert; and (**R4**) $f_x \subseteq f_i$.

R1, R2, and R3 establish that f_i is assigned to s_i, and R4 shows that f_x is a subset of f_i. Suppose f_i is not assigned to s_i but is verified as such (i.e., R1-R4 are met). For this statement to be true, the following statements must be true: $(f_i, s_i)_{s_i}$ is improper; and $(f_i, s_i)_{s_j}$ is improper. Since $(f_i, s_i)_{s_i}$ and $(f_i, s_i)_{s_j}$ are improper and consistent, s_i and s_j either share a common false data source (H1) or they are considered in collusion (H2). R3 reduces the likehood of H1, and H2 is ruled out by assumption A1. Thus, the statement that f_i is not assigned to s_i but is verified as such is, with reasonable assurance, not true. In other words, if f_i is not assigned to s_i, it will, with reasonable assurance, not be verified as such. Equivalently, if f_i is verified as assigned to s_i, it is, with reasonable assurance, assigned to s_i. This establishes Lemma 1.

LEMMA 2. *psBGP provides reasonable assurance of IP prefix aggregation verification.*

Proof Outline: Let f_g be a prefix aggregated by AS s_x from a set of routes $\{m_i{=}(f_i, p_i) | p_i = [s_i, \ldots]\}$ received by s_x. psBGP requires that for f_g originated by s_x to be verified as proper, s_x must either own a prefix f_x such that $f_g \subseteq f_x$ (verified by Lemma 1), or provide evidence that s_x has in fact received $\{m_i\}$ and $f_g \subseteq \cup\{f_i\}$. Valid digital signatures from each AS on p_i can serve as evidence that s_x has received $\{m_i\}$ (see Proposition 4). If $f_g \subseteq \cup\{f_i\}$, then s_x aggregates f_g properly. If s_x cannot provide the required evidence, s_x's aggregation of f_g is verified as improper. This establishes Lemma 2.

PROPOSITION 5. *psBGP provides reasonable assurance of IP prefix origination authentication (G5), i.e., an AS s_i's origination of a prefix f is, with reasonable assurance, verified as proper if f is assigned to s_i or is aggregated properly by s_i from a set of routes received by s_i.*

Proof Outline: Lemma 1 allows prefix assignment verification, and Lemma 2 allows prefix aggregation verification, thus establishing Proposition 5.

The above results (Propositions 1–5) establish the psBGP security properties, as summarized by Theorem 1 (cf. §2.3).

THEOREM 1 (PSBGP SECURITY PROPERTIES). *psBGP achieves the following five security goals: AS number authentication (G1), BGP speaker authentication (G2), data integrity (G3), AS_PATH authentication (G4), and prefix origin authentication (G5).*

5.2 Countering Selected BGP Threats

We first consider how psBGP detects false prefix originations, and next discuss how psBGP reacts to possible new threats arising from proposed security mechanisms in psBGP itself. We also discuss some attack scenarios which are not addressed by psBGP.

5.2.1 *Detecting False Prefix Origin.* We consider three cases in which an AS may originate routes for a prefix which is actually assigned to another AS.

MALICIOUS ATTACK. A malicious AS may hijack a prefix from another AS to attract its traffic. An AS is considered malicious if one or more BGP speakers within that AS are compromised, or the administrator in the AS that controls BGP software and configuration

intentionally misbehaves. psBGP can detect prefix hijacking since a malicious AS will be unable to obtain from its neighbors or a trusted authority (e.g., an RIR) endorsements for the hijacked prefix.

ROUTER MALFUNCTION. A router may mistakenly deaggregate prefixes (e.g., due to software problems) and announce more specific ones. Deaggregating another AS's prefix is referred to as *foreign deaggregation*; deaggregating one's own prefix is referred to as *self deaggregation*. Foreign deaggregation has the same external behavior as prefix hijacking, and thus can be detected. Self deaggregation appears to be equivalent to the announcement of a subset of the prefix assigned to an AS, and thus is treated as legitimate.

DATABASE MISCONFIGURATION. Many ISPs use automatic scripts to generate router configurations from a centralized database containing information of prefix assignments. If a prefix is erroneously entered into such database (e.g., due to human error), automatically generated configurations will instruct a router which might be functioning correctly to originate a prefix which it is not authorized to announce.

Database misconfiguration will not result in successful prefix hijacking if the erroneous database is not used by *any* neighboring AS to generate its PAL. In other words, if the information used by all endorsing ASes for generating $PALs$ is independent of the misconfigured database containing erroneous prefixes, origin of those prefixes will result in verification failures since there will not exist a prefix endorsement consistent with the false prefix assertion. However, an ISP may have multiple ASes and use a single centralized database for generating both router configurations and $PALs$ for its own ASes. Thus, it is possible that an erroneous prefix assertion made by one AS gets endorsement from another AS owned by the same ISP. This scenario is addressed in psBGP with MultiASCerts (Section 4.2). More specifically, an endorsement from s_i for a prefix assertion made by s_j is not used if both s_i and s_j are owned by the same organization, in which case they should both appear on a MultiASCert under a common organization.

5.2.2 *Countering False $PALs$.* We now discuss how psBGP reacts to erroneous $PALs$ that contain false assertions or endorsements. These might potentially introduce new vulnerabilities arising from the proposed enhancements, as a result of malice or human error.

ERRONEOUS PREFIX ASSERTIONS. An AS s_i erroneously asserting the ownership of a prefix through its own PAL will not result in service disruption of the legitimate owner of that prefix as long as none of s_i's neighbors endorses its assertion.

ERRONEOUS PREFIX ENDORSEMENTS. An AS s_i erroneously endorsing s_j for a prefix which is not asserted by s_j will not result in any service disruption since such an endorsement will not be used by any AS when it verifies s_j's prefix assertions. If s_i is the only endorsing neighbor for s_j, or more generally, $\forall s_i \in N(s_j)$, s_i issues $(f'_j, s_j)_{s_i}$ inconsistent with $(f_j, s_j)_{s_j}$, then $(f_j, s_j)_{s_j}$ will be verified as *improper* by other ASes, even if it is actually correct. This is the case when misbehaving ASes form a network cut from s_j to any part of the network. It appears difficult, if not impossible, to counter such an attack; however, we note that even if such a denial of service attack could be prevented, many other techniques beyond the control of BGP could also be used to deny the routing service of s_j, e.g., link-cuts [Bellovin and Gansner 2003], filtering, or packet dropping. Note that a prefix assertion made by s_i about a remote AS s_k, i.e., $s_i \notin N(s_k)$, will not be checked when s_k's prefix assertions are verified because s_i is not a neighbor of s_k. Thus, a misbehaving AS is unable to mislead other ASes about the prefix ownership of a non-neighboring AS.

5.2.3 *Limitations of psBGP.* We now discuss some limitations of psBGP. First, it is subject to human error if a psBGP-enabled AS s_i sets threshold β_i=1 (e.g., during the early stage of psBGP deployment on the Internet). For example, if an AS uses a common database for generating BGP speaker configuration and for issuing $PALs$, a prefix erroneously entered into such a database can result in service disruption. Second, psBGP is subject to k-party collusion if β_i=k≥2. Suppose β_i=2 which is the recommended configuration (see §4.4.1) for each psBGP-enabled AS s_i. If an attacker controls two ASes that are owned by two different organizations (i.e., they do not appear on a common Multi-ASCert), it is possible for the attacker to generate two erroneous yet consistent $PALs$. This is equivalent to the case that the $PALs$ issued by two different ASes are in fact based on a single data source; thus corroborating these two dependent $PALs$ does not yield additional benefit. As a result, psBGP security can be defeated. To successfully launch such an attack, an adversary needs to: a) set up two organizations and manage to obtain an AS number from an RIR for each of them; b) compromise the private keys used by two independent ASes for signing their $PALs$; or c) set up one organization and manage to obtain an AS number from an RIR and compromise the private key used by another AS for signing its PAL. We suggest that these attacks would present non-trivial (albeit not insurmountable) practical difficulty to an adversary.

6. OPERATIONAL ANALYSIS OF PSBGP

Here we analyze some operational and performance issues of psBGP.

6.1 Deployment Analysis of psBGP

We first argue that the effort involved in deploying psBGP is reasonable (relative to alternatives), and next discuss incremental benefits from psBGP deployment.

6.1.1 *Reasonable Deployment Effort.* To deploy psBGP, an AS needs to: upgrade its BGP speakers to support psBGP; issue a single public key certificate for its own BGP speakers (SpeakerCert); distribute the corresponding private key securely to its speakers; and issue an appropriate prefix assertion list (PAL). Upgrading BGP speakers can be done in a similar manner as upgrading existing router software. Issuing a SpeakerCert (e.g., in X.509v3 format) requires some level of knowledge of public key certificates. However, many people responsible for BGP operations might have already acquired similar knowledge, e.g., from the use of PGP [Zsako 1999]; in any case, we acknowledge that additional effort will always be involved in setting up a new system. For example, personnel familiar with PGP may still need to spend some time studying the X.509v3 certificate format. Issuing a PAL requires carrying out a certain level of due diligence in improving an AS' confidence in the prefixes assigned to a (typically) small number of selected neighbors. We expect such effort is reasonable since two direct neighbors usually have established service agreements allowing some level of direct interaction. Such effort is also justifiable (in our opinion) considering potential security benefit to the Internet as a whole. Overall, all of this work can be done independently by an AS without requiring authorization from other ASes (e.g., an upstream ISP). In other words, psBGP can be deployed from the bottom up, mirroring the growth model of the Internet.

6.1.2 *Incremental Deployability.* As with the deployment of almost any other large scale security system, it is unrealistic to expect psBGP to be deployed by all ASes simultaneously, or to be deployed at different times but turned on at the same time. It is expected

that if adopted, a small number of ASes will deploy psBGP first, then more and more ASes will follow. It is desirable that those ASes deploying psBGP first can achieve some immediate benefits to justify their investment before psBGP is widely deployed. Here we analyze benefits and constraints of psBGP deployment (β=1).

The first AS adopting psBGP does not gain any immediate benefit since none of the other ASes speaks psBGP. The second AS adopting psBGP will have some benefit collectively with the first psBGP-enabled AS if they are direct neighbors. In this case, one psBGP-enabled AS (s_i) will likely prefer the route originated by the other (s_j) over routes originated by a non-psBGP enabled AS regarding a prefix assigned to s_j (see §4.5). Since s_i and s_j are also directly connected, traffic originated from s_i and destined to s_j will likely arrive at s_j and not be attracted to another AS if everything else besides BGP also works correctly. In the case that s_i and s_j are not directly connected, i.e., connected by one or more non-psBGP enabled ASes, s_i will still likely prefer the route originated by s_j over an erroneous one by a non-psBGP enabled AS (see §4.5), resulting in containment of any erroneous announcements. However, there is no assurance that traffic destined to s_j can reach their ultimate destinations from s_i. This is because such traffic must traverse through non-psBGP enabled ASes (or unsecured zones), some of which could have poisoned routing tables and direct traffic over incorrect paths. Thus, security that can be achieved by two remote psBGP enabled ASes is less than that achieved by two psBGP-enabled neighbors.

We say that one or more psBGP-enabled ASes with direct links among themselves form a *secure zone*, and one or more non-psBGP enabled ASes with direct links among themselves form a *nonsecure zone*. Assume at one point, there are a number of ASes on the Internet which have deployed psBGP. Then the Internet can be viewed to consist of a number of secure and nonsecure zones. Since two directly connected secure or non-secure zones can always form a larger secure or non-secure zone, a secure zone will always directly connect with nonsecure zones, and a non-secure zone can have only secure zones as its direct zone neighbors. This implies that secure zones can always form a network cut for a nonsecure one. To this end, we can draw two conclusions:

1) An AS improperly originating a route for a prefix assigned to a psBGP-enabled AS will be contained once it reaches a secure zone. In other words, if a misbehaving AS is within a secure zone, the erroneous route will be contained immediately. If it is within a nonsecure zone, it will propagate within the nonsecure zone and be contained once it reaches a secure zone.

2) An improper origination of a prefix assigned to a non-psBGP enabled AS will be propagated (without detection by psBGP) through all non-secure and secure zones, i.e., over the entire Internet.

It is clear from the above conclusions that prefixes assigned to a psBGP-enabled AS are protected to a certain degree from being hijacked while there is no such protection for non-psBGP enabled ASes. While a psBGP-enabled AS might find limited protection when the number of other psBGP-enabled ASes is small, the protection increases as this number grows. As a starting point, it might be beneficial for an organization which owns multiple ASes (such as a large or even medium-sized government) to deploy psBGP so that a secure zone can be formed within that organization.

6.2 Complexity Analysis of psBGP

Here we consider the computational complexity resulting from AS_PATH verification and AS prefix graph related operations. The former involves computationally expensive operations such as digital signature generation and verification, while the latter involves much simpler (less costly but potential numerous) operations such as data structure insertion, deletion, comparison, and query. We do not attempt to provide a detailed, mathematically rigorous running-time analysis for psBGP operations, but rather to provide enough insight to allow ball-bark estimates sufficient to provide confidence that computational costs of psBGP are reasonable, and will not be a reason to avoid deploying psBGP.

6.2.1 *Complexity of AS_PATH Verification.* Let a be the average number of external ASes with which a BGP speaker establishes BGP sessions, and b the average number of ASes on an AS_PATH. A psBGP-enabled BGP speaker needs to generate on average a unique digital signatures (one per AS neighbor) for each BGP update message it sends to a neighbors, and to verify on average b unique digital signatures (for maximal security, i.e., θ=1) for each BGP update message received (see Algorithm 3). Signature verifications related to certificate revocation and certificate chains are ignored here.

6.2.2 *Complexity of AS Prefix Graph Operations.* Let n be the total number of ASes on the Internet, d the average number of AS neighbors, and h the average number of prefixes assigned to an AS. Let $x \leq d$ be the average number of neighboring ASes whose prefix assertions are endorsed by an AS, and y the average number of prefixes endorsed by an AS for each such neighbor. Accordingly, each AS on average has x endorsing neighbors.

Thus, each PAL (cf. §4.1) on average consists of: 1) h prefix assertions, one for each assigned prefix; 2) y prefix endorsements for each endorsed neighbor (x of them), resulting in xy prefix endorsements in total; 3) $d-x$ null prefix endorsements, one for each non-endorsed neighbor. Assume there are z MultiASCerts. We next estimate the computational costs of the construction, update, and query of an AS prefix graph in psBGP. Note all operations mentioned here are simple database operations (e.g., comparison), not computationally expensive operations such as digital signature generation or verification.

1) *Complexity of AS Prefix Graph Construction* (Algorithm 4). For the first pal_i received from each AS on the Internet, an AS needs to update the APAS $H(s_i)$ for s_i (lines 6–13), resulting in $h\{1+d[2+xy(1+z+1+1)]\}$ operations. In addition, an AS also needs to update the APAS $H(s_j)$ for each of s_i's endorsed neighbors s_j (lines 14–20), resulting in $d\{1+h[xy(1+z+1+1)+1]\}$ operations. Thus, in total $2hdxyz+6hdxy+3hd+h+d$ operations are required for processing each pal_i, resulting in $n(2hdxyz+6hdxy+3hd+h+d)$ operations for constructing a complete AS prefix graph from n $PALs$.
2) *Complexity of AS Prefix Graph Update* (Algorithm 5). Consider the worst case that an AS s_i issues a new pal_i' that is completely different from the existing pal_i, i.e., all of its prefix assertions and endorsements have changed. In Algorithm 5, lines 6–7 result in h operations, lines 8–11 result in $5xy$ operations, lines 12–18 result in $5d$ operations, lines 19–25 result in $h\{1+d[xy(1+z+1+1)]+1\}$ operations, and lines 26–31 result in $d\{xy[1+h(1+z+1+1)]\}$ operations. Thus one update might require in total $2hdxyz+6hdxy+dxy+5xy+3h+5d$ operations.
3) *Complexity of AS Prefix Graph Query* (Algorithm 6) When an AS receives a BGP update message, it verifies that the origin AS is allowed to announce the prefix by

comparing the announced prefix with the h prefixes asserted by the origin AS, resulting in up to h operations for one prefix origin verification.

6.3 Performance Analysis of psBGP

Here we present our preliminary estimation of memory, bandwidth, and CPU overhead, and the analysis of certificate dynamics in psBGP. While rigorous study has been performed by Aiello et al. [Aiello et al. 2003] on the prefix delegation stability on the Internet as a whole, and by Zhao et al. [Nicol et al. 2004; Zhao et al. 005a] on PKI impact on BGP security using simulation, it is desirable to study certificate dynamics of a secure system and to project certificate management overhead on a per-AS level. We use BGP data collected by the RouteViews project [RouteViews 2005]. We retrieved one BGP routing table the first day of each month from January to August 2004. Despite known shortcomings including incompleteness of the RouteViews data set, it is one of the most complete data repositories publicly available, and has been widely used in the BGP community.

6.3.1 *Memory Overhead.* Four types of certificates and one AS prefix graph require memory for a BGP speaker to support psBGP. We estimate the memory overhead for each type and then give an estimate of the total. While a BGP update message may carry extra digitally signed data and signatures which need to be stored temporarily, they can be discarded after verification. Thus, we omit their memory overhead here.

ASNUMCERTS AND SPEAKERCERTS. We observed in total $17\,884$[4] ASes as of August 1, 2004. One ASNumCert is required per AS. In the worst case, an AS may need to store the ASNumCert of every AS on the Internet; in this case, $17\,844$ ASNumCerts would be stored. As with S-BGP and soBGP, psBGP recommends use of the X.509v3 certificate structure which has wide industrial support. Assuming the average size of a certificate is 600 bytes [Kent 2003] based on 1024-bit RSA keys, 10.479M bytes of memory would be required for storing $17\,844$ ASNumCerts. The same holds for SpeakerCerts.

$PALs$ AND MULTIASCERTS. The size of pal_i, issued by each AS s_i, is primarily determined by the number of prefixes assigned to s_i, the number of s_i's neighbors, and the number of prefixes assigned to each of s_i's neighbors that are endorsed by s_i. While some ASes have many neighbors, and some are delegated many prefixes, many ASes have only a small number of neighbors and are delegated a small number of prefixes. Based on the RouteViews data we use, each AS on average has 4.2 neighbors and is delegated 9.1 prefixes. Assuming the average size of a PAL is 1024 bytes (600 bytes for an X.509v3 certificate plus 424 bytes for about 60 prefix assertions and endorsements), 17.844M bytes of memory would be required to store $17\,844$ PALs, one for each AS. For MultiASCerts, a BGP speaker needs to store one certificate for each organization which owns multiple ASes. Based on the data from Aiello et al. [Aiello et al. 2003], there are 385 multi-AS organizations which in total own 1259 ASes. On average, each multi-AS organization owns 3.3 ASes. Assuming the average size of a MultiASCert is 600 bytes, 0.226M bytes of memory are required by each AS for storing all MultiASCerts.

AS PREFIX GRAPH. Each AS needs to construct an AS prefix graph for prefix origin verification. The memory space required for storing an AS prefix graph depends on the data structures being used. For simplicity, we use a fixed array consisting of $17\,844$ entries, one entry per AS. Each entry consists of a 16-bit AS number and two 32-bit pointers, one

[4] AS numbers used by IANA itself for experimental purposes are not counted.

pointing to a linked list of prefixes assigned to this AS and the other pointing to a linked list of neighboring ASes. On average, each prefix linked list has 10 elements with each of 17 bytes and each neighboring AS linked list has 5 elements with each of 6 bytes. Thus, each entry in the fixed array on average consumes 210 bytes. In total, an AS prefix graph requires 3.747M bytes memory (M=10^6), using these (non-optimized) data structures.

ASNumCerts	10.479M Bytes
SpeakerCerts	10.479M Bytes
PALs	17.844M Bytes
MultiASCerts	0.226M Bytes
AS Prefix Graph	3.747M Bytes
Total	41.775M Bytes

Table III. psBGP Memory Requirements per AS

In summary, a total of 41.775M bytes of memory are required for storing all certificates and an AS prefix graph to support psBGP (see Table III).

6.3.2 *Bandwidth Overhead.* Except for a small number of public key certificates of trusted CAs which may be distributed using out-of-band mechanisms, all other certificates in psBGP can be distributed with BGP update messages. The latter consumes extra network bandwidth. However, such overhead is not persistent since those certificates only need to be distributed periodically or upon changes. We expect that such overhead is of little significance and do not discuss it further.

The primary bandwidth overhead is introduced by digitally signed data and signatures carried by each BGP update message for protecting the message. For a fully protected BGP route where every AS on the route digitally signs the update message, the overhead is mainly determined by the number of such ASes, and could result in as much as 600% overhead according to Kent [Kent 2003]. We expect no significant difference between the bandwidth overhead of psBGP and S-BGP. While increased bandwidth overhead due to psBGP (or e.g., S-BGP) is significant in terms of percentage, as pointed out by Kent [Kent 2003], BGP control messages only account for a small fraction of network bandwidth versus subscriber traffic. Thus, from our preliminary analysis, we expect that bandwidth overhead of psBGP will not create difficulty in the deployment of psBGP.

6.3.3 *CPU Overhead.* We expect that CPU overhead of psBGP will mainly result from AS_PATH verification, not AS prefix graph operations (cf. §6.2). A psBGP-enabled BGP speaker needs to digitally sign each BGP update message sent to each neighbor, and to verify some digital signatures carried by each BGP update message it receives and chooses to use. As shown by Kent et al. [Kent et al. 2000] in their study of S-BGP performance, such CPU overhead is significant. Especially in the case of reboots, a BGP speaker will receive full routing tables from each of its neighbors, and thus must verify a large number of digital signatures if psBGP is implemented. Note an AS prefix graph need not be rebuilt since it can be stored in persistent storage and reloaded upon reboot. psBGP provides the flexibility for reducing the CPU overhead resulting from digital signature verification by using a lower confidence threshold, which trades off security for efficiency. In other words, psBGP provides a mechanism which allows protection to be proportionally achieved in accordance to the CPU power which a router has available to spend on signature verification.

However, to achieve higher level of assurance of AS_PATH integrity, significant CPU overhead will be generated by psBGP. To mitigate the problem, various approaches might be helpful, including caching [Kent et al. 2000], delay of signature verification [Kent et al. 2000], using a digital signature algorithm with a faster verification operation (e.g., RSA) [Nicol et al. 2004], and aggregated path authentication [Zhao et al. 2005].

6.3.4 *Certificate Dynamics.* ASNUMCERTS AND SPEAKERCERTS. The monthly number of ASes on the Internet has grown by an average of 190 since January 1, 2004, with an average of 347 ASes added and 157 ASes removed (see Table II). When an AS number is added or removed in psBGP, the corresponding ASNumCert must be issued or revoked by an RIR. Thus, four RIRs between them must issue an average of 347 new ASNumCerts and revoke an average of 157 existing ASNumCerts per month. This would certainly appear to be manageable in light of substantially larger PKIs existing in practice (e.g., see [Guida et al. 2004]). Note the issuing and revocation of a SpeakerCert is performed by an AS, not an RIR.

PREFIX ASSERTION LISTS (PALS). A prefix assertion list pal_i must be changed (removed, added, or updated) if: a) the AS number s_i changes (i.e., is removed or added); b) an IP prefix assigned to s_i changes; c) s_i's neighbor relationship changes, i.e., a neighbor is removed or added; or d) an IP prefix changes which is endorsed by s_i for one of its neighbors. Table IV depicts the dynamics of prefix assignments.

	Jan	Feb	Mar	Apr	May	Jun	Jul
Start of Month	148 903	148 014	151 174	156 019	157 925	160 818	155 118
Stable During Month	143 200	144 422	146 139	151 481	153 171	148 280	151 436
Stable During Jan-Jul	119 968	119 968	119 968	119 968	119 968	119 968	119 968
Removed During Month	5 703	3 592	5 035	4 538	4 754	12 538	3 682
Added During Month	4 814	6 752	9 880	6 444	7 647	6 838	10 360

Table IV. IP Prefix Dynamics from January 1 to August 1, 2004

We study the number of prefix assertion (PA) changes required for each AS based on the two routing tables of July 1 and August 1, 2004. Each prefix addition or removal is counted once (i.e., resulting in one PA addition or removal) if the AS number of the AS owning that prefix does not change. If an AS number is newly added (or removed) during the month, all additions (or removals) of the prefixes owned by that AS are counted once as a whole. One PA change usually represents one update to a PAL if such update is done in a timely manner. However, an AS can choose to do multiple PA changes in one PAL update (see §6.4 for more discussions).

Table V depicts the projected PA dynamics based on the data set of July 2004. The total number of ASes observed during July 2004 is $18\,048$, including $17\,884$ observed on August 1, 2004 and 164 removed during July 2004. We can see, the more ASes endorsing an AS's prefix assertions, the more PA changes required. We recommend the scenario $n = 2$, where each AS has at most two endorsing neighbors even if it has more than two neighbors. This provides a level of redundancy in the case that one of the two endorsing neighbors fails to carry out adequate due diligence.

From Table V, in the recommended scenario $n = 2$, 16% of the ASes need to update their $PALs$ during the month. 8.4% of ASes need only one PA change in the month, 4% need 2 to 4 PA changes, and 1.9% need 5 to 10 PA changes. However, a small number of

# of PA Changes		1	2-4	5-10	11-30	31-100	101-1000	over 1001	Total
n=1	# of ASes	1497	677	319	152	69	26	2	2742
	(percentage)	(8.3%)	(3.8%)	(1.8%)	(0.8%)	(0.3%)	(0.1%)	(0%)	(15.2%)
n=2	**# of ASes**	**1508**	**713**	**346**	**187**	**87**	**48**	**3**	**2892**
	(percentage)	**(8.4%)**	**(4.0%)**	**(1.9%)**	**(1.0%)**	**(0.5%)**	**(0.2%)**	**(0%)**	**(16.0%)**
n=3	# of ASes	1516	725	355	205	93	54	4	2952
	(percentage)	(8.4%)	(4.0%)	(2.0%)	(1.1%)	(0.5%)	(0.3%)	(0%)	(16.4%)
n=all	# of ASes	1424	784	387	233	112	53	30	3023
	(percentage)	(7.9%)	(4.3%)	(2.1%)	(1.3%)	(0.6%)	(0.3%)	(0.2%)	(16.7%)

Table V. Projected number of ASes in absolute number, and as percentage of all ASes, requiring the specified number of monthly prefix assertion (PA) changes in psBGP based on July 2004 data. We recommend row $n = 2$ (n is the number of endorsing neighbors).

ASes need more than 100 changes, and AS 701 (UUNET) and its two endorsing neighbors need around 5000 changes. In our study, if an AS chooses to endorse the prefixes of a neighboring AS, it simply endorses all the prefixes assigned to that neighbor. To reduce the number of PA changes, an AS can choose to only endorse a subset of the prefixes assigned to a neighbor. In this case, PA change overhead can be distributed to some other ASes and will be more balanced than what is shown in Table V.

6.4 Discussion

The timeliness of PAL updates is important to ensure service availability. $PALs$ need to be updated and distributed in a timely manner so that prefix ownerships can be verified using currently correct information. To ensure that an endorsing neighbor of a given AS updates its PALs for that AS in a timely manner, a service agreement between them would likely be required, e.g., as an extension to their existing agreements. Since there is usually some time delay window before newly delegated prefixes are actually used on the Internet, an endorsing AS should be required to update its PAL to include newly delegated prefixes of an endorsed neighbor within that delay window. Updates of prefix removals can be done with lower priority since they would appear to have only relatively small security implications. PALs along with other certificates (e.g., ASNumCerts, SpeakerCerts, and corresponding Certificate Revocation Lists) can be distributed with BGP update messages in newly defined path attributes [Kent 2003]; thus, they can be distributed as fast as announcements of prefixes and are accessible without any dependence on BGP routes. Those certificates might also be stored in centralized directories [Kent 2003]. However, a "pull" model might make it challenging to decide how often centralized directories should be checked.

7. RELATED WORK

Considerable research has been published on securing routing protocols. Perlman [Perlman 1988] was among the first to recognize and study the problem of securing routing infrastructures. Bellovin [Bellovin 1989] discussed security vulnerabilities of Internet routing protocols as early as 1989 (see also [Bellovin 2004]). More recently, Bellovin and Gansner [Bellovin and Gansner 2003] discussed potential link-cutting attacks against Internet routing. Kumar [Kumar and Crowcroft 1993] proposed the use of digital signatures and sequence numbers for protecting the integrity and freshness of routing updates. Smith et

al. [Smith and Garcia-Luna-Aceves 1996] proposed the use of digital signatures, sequence numbers, and a loop-free path finding algorithm for securing distance vector routing protocols including BGP. For a thorough analysis of BGP vulnerabilities and protections, see Murphy [Murphy 2002b; 2002a].

The most complete and concrete security proposal to date for addressing BGP vulnerabilities is S-BGP [Kent et al. 2000; Kent et al. 2000; Seo et al. 2001]. It proposes the use of centralized PKIs for authenticating AS numbers and IP prefix ownership. S-BGP PKIs are rooted at RIRs, and parallel to the existing system of AS number assignment and IP address allocation. AS_PATH is protected using nested digital signatures, and the integrity of an AS_PATH is guaranteed.

soBGP [White 2003] proposes the use of a web-of-trust model for AS public key authentication, and a centralized hierarchical model for IP prefix ownership verification. AS_PATH is verified for plausibility by checking against an AS topology graph. Each AS issues certificates listing all peering ASes. A global AS graph can be constructed from those certificates. Thus, the existence of an AS_PATH can be verified. Table VI compares S-BGP, soBGP, and psBGP (recall §2.3 re: goals, also see §3.5 and [Wan et al. 2005] for further background information).

Goal	S-BGP	soBGP	psBGP
G1: AS Number Authentication	centralized (multiple levels)	decentralized (with trust transitivity)	centralized (depth=1)
G2: BGP Speaker Authentication	one certificate per BGP speaker	one certificate per AS	one certificate per AS
G3: Data Integrity	IPsec or TCP MD5	IPsec or TCP MD5	IPsec or TCP MD5
G4: Prefix Origination Verification	centralized (multiple levels)	centralized (multiple levels)	decentralized (no trust transitivity)
G5: AS_PATH Verification	full integrity	plausibility	stepwise integrity

Table VI. Comparison of S-BGP, soBGP, and psBGP re: achieving security goals of §2.3.

Goodell et al. [Goodell et al. 2003] proposed a protocol, namely Inter-domain Routing Validator (IRV), for improving the security and accuracy of BGP. Each AS builds an IRV server which is authoritative of the inter-domain routing information of that AS. An IRV can query another IRV to verify BGP update messages received by its hosting AS. Improper prefix origination and AS_PATH might be detected by uncovering the inconsistency among responses from other IRVs. One advantage of IRV is that it supports incremental deployment since it does not require changes to the existing routing infrastructure.

Kruegel et al. [Kruegel et al. 2003] propose a model of AS topology augmented with physical Internet connectivity to detect and stop anomalous route announcements. Their approach passively monitors BGP control traffic, and does not require modification to the existing routing infrastructure. Therefore, it would appear to be easy to deploy.

In a rigorous study of prefix origination authentication, Aiello et al. [Aiello et al. 2003] formalize the IP prefix delegation system, present a proof system, and propose efficient constructions for authenticating prefix origination. Real routing information is analyzed and used to reconstruct the IP delegation relationship over the Internet. They discover that the current prefix delegation on the Internet is relatively static and dense, however they also note that it is extremely difficult, if not impossible, to determine this delegation structure.

Listen and Whisper [Subramanian et al. 2004] are proposed mechanisms for protecting the BGP data plane and control plane respectively; they are best used together. The first approach (Listen) detects invalid data forwarding by detecting "incomplete" (as defined in [Subramanian et al. 2004]) TCP connections. Whisper uncovers invalid routing announcements by detecting inconsistency among *path signatures* of multiple update messages, originating from a common AS but traversing different paths.

Hu et al. [Hu et al. 2004] propose a Secure Path Vector (SPV) protocol for securing BGP. SPV makes use of efficient cryptographic primitives, e.g., authentication trees and one-way hash chains for protecting AS_PATH, and is argued being more efficient than S-BGP.

8. CONCLUDING REMARKS

Different approaches have been taken by S-BGP and soBGP for addressing security in BGP. We believe that psBGP adopts their best features, while differing fundamentally with a novel approach taken to verify IP prefix assignments and AS_PATH integrity. As no centralized infrastructure for tracing changes in IP prefix assignments currently exists, and it would appear to be quite difficult to build such an infrastructure, we believe that the decentralized approach taken by psBGP provides a more feasible means of increasing confidence in correct prefix origin.

Beyond AS_PATH verification in §3.5, it is desirable to verify if an AS_PATH conforms to the route exporting policies of each AS on the path. Since BGP is a policy-driven routing protocol, each AS can individually decide whether or not a received route advertisement should be further propagated to a neighboring AS. Such route exporting policies are mainly defined based on the business relationship with a neighboring AS. Without such verification, a misbehaving BGP speaker (e.g., misconfigured) may be able to re-advertise routes which are prohibited by its route exporting policies. For example, a multihomed AS may readvertise routes received from one provider AS to the other, thus functioning as a transit AS for its two providers. Such misbehavior may allow for eavesdropping and may also result in service disruption. We are currently exploring new mechanisms for AS_PATH verification, which we expect to present in future work.

Finally, we hope that this paper will serve to stimulate discussion in the Internet community about alternate design choices and trust models for securing BGP.

Acknowledgements

We thank Steve Bellovin, Stephen Chou, John Ioannidis, Angelos Keromytis, Stefan Mink, and anonymous reviewers for their constructive comments on preliminary paper [Wan et al. 2005] and related discussions; these significantly improved the advancement of psBGP, and the present paper. We also specifically thank Steve Bellovin for pointing out the collusion problem of multi-AS organizations and for motivating the proposal as described in §4.2. The first author is supported in part by MITACS (Mathematics of Information Technology and Complex Systems) and NSERC (Natural Sciences and Engineering Research Council of Canada). The second author is Canada Research Chair in Network and Software Security, and is supported in part by NCIT (National Capital Institute of Telecommunications), MITACS, an NSERC Discovery Grant, and the Canada Research Chairs Program. The third author is supported in part by Alcatel Canada, MITACS and NCIT.

Version: September 20, 2005.

REFERENCES

ADAMS, C. AND LLOYD, S. 2003. *Understanding Public-Key Infrastructure, 2^{nd} edition*. Addison-Wesley Professional.

AIELLO, W., IOANNIDIS, J., AND MCDANIEL, P. 2003. Origin Authentication in Interdomain Routing. In *Proc. of 10^{th} ACM Conference on Computer and Communications Security*. Washington, D.C., USA, 165–178.

ATKINSON, R. AND FLOYD, S. 2004. IAB Concerns and Recommendations Regarding Internet Research and Evolution. IETF RFC 3869.

BARBIR, A., MURPHY, S., AND YANG, Y. 2004. Generic Threats to Routing Protocols. Internet Draft.

BELLOVIN, S. 1989. Security Problems in the TCP/IP Protocol Suite. In *Computer Communications Review*. Vol. 19. 32–48.

BELLOVIN, S. 2004. A Look Back at "Security Problems in the TCP/IP Protocol Suite". In *The 20^{th} Annual Computer Security Applications Conference (ACSAC'04)*. Tucson, Arizona, USA.

BELLOVIN, S. AND GANSNER, E. 2003. Using Link Cuts to Attack Internet Routing. Unpublished manuscript.

BELLOVIN, S., IOANNIDIS, J., AND BUSH, R. 2005. Position Paper: Operational Requirements for Secured BGP. In *DHS Secure Routing Workshop*.

BONEH, D., BOYEN, X., AND SHACHAM, H. 2004. Short Group Signatures. In *Proceedings of Crypto 2004*. Vol. 3152. 41–55.

BURROWS, M., ABADI, M., AND NEEDHAM, R. 1989. A Logic of Authentication. In *Research Report 39*. Digital Systems Research Center.

DEMPSTER, A. 1967. Upper and Lower Probabilities Induced by a Multivalued Mapping. *28*, 325–339.

GAARDER, K. AND SNEKKENES, E. 1991. Applying a Formal Analysis Technique to the CCIT X.509 Strong Two-Way Authentication Protocol. *Journal of Cryptology 3*, 81–98.

GAO, L. 2000. Inferring Autonomous System Relationships in the Internet. In *IEEE Global Internet*.

GLIGOR, V., KAILAR, R., STUBBLEBINE, S., AND GONG, L. 1991. Logics for cryptographic protocols - virtues and limitations. In *Proceedings of the Computer Security Foundations Workshop IV*. Los Alamitos, California, USA, 219–226.

GOODELL, G., AIELLO, W., GRIFFIN, T., IOANNIDIS, J., MCDANIEL, P., AND RUBIN, A. 2003. Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing. In *Proceedings of the 2003 ISOC Symposium on Network and Distributed Systems Security (NDSS'03)*. San Diego, California, USA, 75–85.

GUIDA, R., STAHL, R., BUNT, T., SECREST, G., AND MOORCONES, J. 2004. Deploying and Using Public Key Technology: Lessons Learned in Real Life. *IEEE Security and Privacy*, 67–71.

HEDRICK, C. 1988. Routing information protocol. IETF RFC 1058.

HEFFERNAN, A. 1998. Protection of BGP sessions via the TCP MD5 signature option. IETF RFC 2385.

HU, Y., PERRIG, A., AND SIRBU, M. 2004. SPV: Secure Path Vector Routing for Securing BGP. In *Proceedings of ACM 2004 SIGCOMM*. Portland, OR, USA.

IRR. 2005. Internet Routing Registry. http://www.irr.net.

JUST, M., KRANAKIS, E., AND WAN, T. 2003. Resisting Malicious Packet Dropping in Wireless Ad Hoc Networks. In *Proceedings of the 2^{nd} Annual Conference on Adhoc Networks and Wireless (ADHOCNOW'03)*. Montreal, Canada.

KENT, S. 2003. Securing the Border Gateway Protocol: A Status Update. In *Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*.

KENT, S. AND ATKINSON, P. 1998a. Security Architecture for the Internet Protocol. IETF RFC 2401.

KENT, S. AND ATKINSON, P. 1998b. IP Encapsulating Security Payload (ESP). IETF RFC 2406.

KENT, S., LYNN, C., MIKKELSON, J., AND SEO, K. 2000. Secure Border Gateway Protocol (S-BGP) Real World Performance and Deployment Issues. In *Proceedings of the 2000 ISOC Symposium on Network and Distributed Systems Security (NDSS'00)*.

KENT, S., LYNN, C., AND SEO, K. 2000. Secure Border Gateway Protocol (S-BGP). *IEEE Journal on Selected Areas in Communications 18*, 4 (April).

KOBLITZ, N. AND MENEZES, A. 2004. Another Look at "Provable Security". Cryptology ePrint Archive, Report 2004/152. To Appear in Journal of Cryptology. http://eprint.iacr.org/2004/152/.

KRUEGEL, C., MUTZ, D., ROBERTSON, W., AND VALEUR, F. 2003. Topology-Based Detection of Anomalous BGP Messages. In *Proceedings of the 6^{th} Symposium on Recent Advances in Intrusion Detection (RAID'03)*.

KUMAR, B. AND CROWCROFT, J. 1993. Integrating Security in Inter-domain Routing Protocols. *ACM SIGCOMM Computer Communication Review 23*, 5 (October), 36–51.

LYNN, C., KENT, S., AND SEO, K. 2003. X.509 Extensions for IP Addresses and AS Identifiers. draft-ietf-pkix-x509-ipaddr-as-extn-02.txt.

MAURER, U. 1996. Modelling a Public-Key Infrastructure. In *Proceedings of the 4^{th} European Symposium on Research in Computer Security (ESORICS'96)*. 324–350.

MURPHY, S. 2002a. BGP Security Protections. draft-murphy-bgp-protect-00.txt.

MURPHY, S. 2002b. BGP Security Vulnerabilities Analysis. draft-murphy-bgp-vuln-00.txt.

NICOL, D., SMITH, S., AND ZHAO, M. 2004. Evaluation of efficient security for BGP route announcements using parallel simulation. *Simulation Pratice and Theory Journal, special issue on Modeling*.

PERLMAN, R. 1988. Network Layer Protocols with Byzantine Robustness. Tech. Rep. MIT/LCS/TR-429. October.

REITER, M. AND STUBBLEBINE, S. 1997. Toward Acceptable Metrics of Authentication. In *Proceedings of 1997 IEEE Symposium on Security and Privacy*. 10–20.

REKHTER, Y. AND LI, T. 1995. A Border Gateway Protocol 4 (BGP 4). IETF RFC 1771.

RETANA, A. AND WHITE, R. 2002. BGP Custom Decision Process. Internet Draft.

ROUTEVIEWS. 2005. Route Views Project. http://www.routeviews.org.

SEO, K., LYNN, C., AND KENT, S. 2001. Public-Key Infrastructure for the Secure Border Gateway Protocol (S-BGP). In *IEEE DARPA Information Survivability Conference and Exposition II*.

SHAFER, G. 1976. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, USA.

SMITH, B. AND GARCIA-LUNA-ACEVES, J. 1996. Securing the Border Gateway Routing Protocol. In *Proceedings of Global Internet'96*. London, UK, 20–21.

SUBRAMANIAN, L., ROTH, V., STOICA, I., SHENKER, S., AND KATZ, R. 2004. Listen and Whisper: Security Mechanisms for BGP. In *Proceedings of the First Symposium on Networked Systems Design and Implementation (NSDI'04), San Francisco, CA, USA*.

VILLAMIZAR, C., ALAETTINOGLU, C., MEYER, D., AND MURPHY, S. 1999. Routing Policy System Security. IETF RFC 2725.

WAN, T., KRANAKIS, E., AND VAN OORSCHOT, P. 2004. S-RIP: A Secure Distance Vector Routing Protocol. In *Proc. of Applied Cryptography and Network Security (ACNS'04), (academic track)*. Vol. 3089. Yellow Mountain, China, 103–119.

WAN, T., KRANAKIS, E., AND VAN OORSCHOT, P. 2005. Pretty Secure BGP (psBGP). In *Proceedings of the 2005 ISOC Symposium on Network and Distributed Systems Security (NDSS'05)*. San Diego, California, USA.

WAN, T., VAN OORSCHOT, P., AND KRANAKIS, E. 2005. A Selective Introduction to Border Gateway Protocol (BGP) Security Issues. In *Proc. of NATO Advanced Studies Institute on Network Security and Intrusion Detection*. Nork, Yerevan, Armenia. IOS Press (to appear, 2006).

WHITE, R. 2003. Securing BGP Through Secure Origin BGP. *The Internet Protocol Journal 6*, 3 (September), 15–22.

WHITE, R., MCPHERSON, D., AND SANGLI, S. 2004. *Practical BGP*. Addison-Wesley, Reading, MA.

ZHAO, M., SMITH, S., AND NICOL, D. 2005. Aggregated Path Authentciation for Efficient BGP Security. In *Proceedings of 12^{th} ACM Conference on Computer and Communications Security*. Alexandria, VA, USA.

ZHAO, M., SMITH, S., AND NICOL, D. 2005a. Evaluating the Performance Impact of PKI on BGP Security. In *Proceedings of 4^{th} Annual PKI Research Workshop (PKI'05)*. Gaithersburg, MA, USA.

ZIMMERMANN, P. 1995. *The Official PGP User's Guide (second printing)*. Cambridge, MA: MIT Press.

ZSAKO, J. 1999. PGP Authentication for RIPE Database Updates. IETF RFC 2726.