Sign What You Really Care About – Secure BGP AS Paths Efficiently *

Yang Xiang[1,3], Zhiliang Wang[2,3],
Jianping Wu[1,2,3], Xingang Shi[2,3], and Xia Yin[1,3]

[1] Tsinghua National Laboratory for Information Science and Technology (TNList)
[2] Department of Computer Science & Technology, Tsinghua University
[3] Network Research Center, Tsinghua University, Beijing, P. R. China, 100084
{xiangy08, wzl, jianping, shixg, yxia}@csnet1.cs.tsinghua.edu.cn

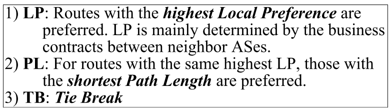
Abstract. The inter-domain routing protocol, Border Gateway Protocol (BGP), plays a critical role in the reliability of the Internet routing system, but forged routes generated by malicious attacks or mis-configurations may devastate the system. The security problem of BGP has attracted considerable attention, and although several solutions have been proposed, none of them have been widely deployed due to weaknesses such as high computational cost or potential security compromise. This paper proposes Fast Secure BGP (FS-BGP), an efficient mechanism for securing AS paths and preventing prefix hijacking by signing critical AS path segments. We prove that FS-BGP can achieve a similar level of security as S-BGP, but with much higher efficiency. Our experiments use BGP UPDATE data collected from real backbone routers. Compared with S-BGP, FS-BGP only requires a very small cache, and can reduce the cost of signing and verification by orders of magnitude. Indeed, the signing and verification can be accomplished as fast as the most bursty BGP UPDATE arrivals, which implies that FS-BGP will hardly delay the propagation of routing information.

Keywords: Inter-Domain Routing, BGP, Prefix Hijacking, Security

1 Introduction

As BGP [13] controls the packet forwarding path between Autonomous Systems (ASes), it plays a critical role in the reliability of the Internet. However, routing information received from neighbors can not be validated. Forged routes may cause packets being forwarded along wrong paths. Malicious attacks often use BGP prefix hijacking to drop, intercept or tamper traffic towards specific prefixes. In 2008, US DoD networks were hijacked at least 7 times [21]. Accidental mis-configurations have also resulted in serious routing problems and economic

* This work is supported by (1) the National Key Technology R&D Program of China under Grant No. 2008BAH37B03, and (2) the National Basic Research Program of China (973 Program) under Grant No. 2009CB320502.



1) **LP**: Routes with the ***highest Local Preference*** are preferred. LP is mainly determined by the business contracts between neighbor ASes.
2) **PL**: For routes with the same highest LP, those with the ***shortest Path Length*** are preferred.
3) **TB**: ***Tie Break***

Fig. 1. Decision process in BGP.

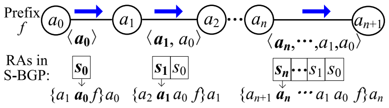

Prefix
f
a_0
a_1
a_2
a_n
a_{n+1}
$\langle \boldsymbol{a_0} \rangle$
$\langle \boldsymbol{a_1}, a_0 \rangle$
$\langle \boldsymbol{a_n}, \cdots, a_1, a_0 \rangle$
RAs in
S-BGP:
$\boldsymbol{s_0}$
$\boldsymbol{s_1}$ s_0
$\boldsymbol{s_n}$ $\cdots$ s_1 s_0
$\{a_1\ \boldsymbol{a_0}\ f\}a_0$
$\{a_2\ \boldsymbol{a_1}\ a_0\ f\}a_1$
$\{a_{n+1}\ \boldsymbol{a_n}\ \cdots\ a_1\ a_0\ f\}a_n$

Fig. 2. Route Attestations (RAs) in S-BGP

loss. For instance, in 2008, Pakistan Telecom hijacked YouTube's prefixes and knocked it off the Internet for two hours [14].

Several extensions have been proposed to improve the security of BGP, including S-BGP [9] and many others. However, S-BGP consumes significant resources of computation and storage, and also faces the problem of replay attack. The other solutions either compromise in the security [6, 19, 7], or bring in more complexity on message size and certification distribution [10].

Towards these unsolved issues, we propose an efficient approach, FS-BGP (Fast Secure BGP), to secure AS paths and prevent prefix hijacking. Through signing critical AS path segments (i.e., adjacent AS triples), FS-BGP can achieve a similar level of security as S-BGP. We evaluate the performance of FS-BGP by BGP UPDATE collected from real backbone routers. Even using a small cache, the signing and verification overhead of FS-BGP account for only 0.56% and 3.9% of that of S-BGP respectively. Indeed, signing and verification can always be accomplished as fast as the most bursty BGP UPDATE arrivals.

This paper is organized as follows. Section 2 introduces backgrounds. Section 3 illustrates our key observation. Section 4 presents the design of FS-BGP and proves its security guarantees. Section 5 reviews the security level of FS-BGP, and section 6 evaluates its performance. Section 7 discusses further extensions including multiple prefixes and complex policies. Finally section 8 concludes.

2 Backgrounds

2.1 BGP and S-BGP

We model the inter-domain routing system as an AS graph, where each AS is denoted by its AS number (ASN). ASes sharing a common edge are called *neighbors*. We are mostly concerned the AS path $p = \langle a_n, \cdots, a_0 \rangle$ embedded in a BGP UPDATE, where the last AS a_0 is the *origin* AS of the path.

BGP is a policy-based routing protocol. An AS only exports a route [4] to a neighbor if it is willing to forward traffic to the corresponding prefix from that neighbor. If an AS receives multiple routes to the same prefix, it chooses and announces the best one according to the decision process as shown in Fig. 1.

In BGP, neither the origin AS nor the AS path is guaranteed to be correct. Secure BGP (S-BGP) [9] is the dominant solution to this problem. S-BGP uses Address Attestations (AAs) for origin authentication, and Route Attestations (RAs) for path authentication. As shown in Fig. 2, an RA is a signature signed by an AS to authenticate the existence and position of ASes in the path. We

[4] We will use route and path interchangeably in this paper.

define $\{msg\}a_i$ as the signature on msg generated with AS a_i's private key. In Fig. 2, each AS a_i signs the corresponding path $\langle a_{i+1}, a_i, \cdots, a_0\rangle$ and the prefix f. The inclusion of the recipient AS a_{i+1} in each signature is necessary to prevent *cut-and-paste* attacks.

S-BGP can protect the network against fabricated routing information, but not against some inside attacks. For instances, a malicious AS can (1) re-announce signed but outdated routes, (2) violate its routing policy and announces routes received from one provider to other providers [5], (3) hijack a prefix through link-cut [3], and (4) selectively drop updates or announce a false withdraw. However, completely securing BGP from inside attacks is difficult. Although S-BGP is not omnipotent, we regard it as currently the most secure scheme with enough capabilities [5], and aim to provide as similar level of security guarantee as S-BGP.

2.2 Related Works

The main concerns about deploying S-BGP in practice include difficulties in maintaining the Public Key Infrastructure (PKI), and the huge computational cost for signing and verifying signatures. Some solutions try to replace PKI but can not guarantee security [12], and some can not provide real-time protection [8, 16]. Since PKI has been adopted by the IETF, and regional registries have already started offering related services [15], we believe PKI is an essential part in the BGP security framework, and use it as our basic building block.

On the other hand, since the number of prefixes is much smaller than the number of paths, and most prefix ownerships are quite stable, AAs can be signed and verified out-of-band. Therefore, the dominating barrier for adopting S-BGP is the overhead of processing RAs, that is to authenticate paths.

Toward this direction, there are a bunch of solutions for reducing the overhead of path authentication. SoBGP [19] maintains all authenticated AS edges in a database, but faces the problem of forged paths. IRV [6] builds an authentication server in each AS, but brings the problem of maintaining and inter-connecting these servers, and introduces query latencies. SPV [7] accelerates the signing process by pre-generated one-time signatures based on a root value, but involves a significant amount of state information, and its security can only be guaranteed probabilistically. Signature Amortization [10] uses a bit-vector to indicate the allowed recipients of a route, such that only one signing is needed for all neighbor recipients. However, each AS will need to pre-establish a neighbor list corresponding to the bit vector, and to distribute it to all other ASes.

As we can see, existing methods for alternating S-BGP usually compromise security, or only improve the performance of signing. However, verification happens more frequently, since one signature needs to be verified at multiple places.

Scope of this paper Accordingly, it is important to design an efficient method to secure AS paths. Our solution, FS-BGP, builds on the assumption that a PKI is ready for use, and focuses on AS path authentication. For origin authentication, FS-BGP uses the same mechanism as S-BGP.

3 Key Observations

Dilemma of S-BGP S-BGP's intention of signing every AS path is reasonable, but it is not realistic because of the high computational cost. And more importantly, it's not worth the cost. BGP is a policy-based routing protocol. An AS only exports a route to a neighbor if it is willing to forward traffic from that neighbor. Under a stable AS-level topology, we call a path *available* when the path satisfies the *import and export policies* of all ASes along the path. We further divide all available paths into three categories, according to the decision process of BGP as shown in Fig. 1:

- *Optimal path*: the best path that passes all the three decision steps.
- *Sub-optimal path*: paths with the same Local Preference as the optimal path, but not chosen as the best one.
- *Suppressed path*: paths with lower LP than the optimal path. Expensive paths (i.e., through a provider) are often suppressed by a low LP.

BGP only announces one available path for every prefix each time. However, since failures occur quite frequently in the global routing system, sub-optimal and suppressed path can be easily announced and propagated. For this reason, S-BGP actually authenticates all announced available paths. In extreme cases, S-BGP even authenticates almost *all available paths*. On the other hand, S-BGP can not prevent replay of non-optimal paths. It can only use the expiration-date (up to several days) to roughly control the window exposed to a replay attack.

NBIE Although complex policies (i.e., route filters [2]) exist, an AS usually does not differentiate those nonadjacent ASes. For example, in Fig. 3, when a_n decides whether routes learned from a_{n-1} can be exported to a_{n+1}, it only considers its relation with the two neighbors (i.e., business partners), but does not consider other ASes along the path (i.e., $a_{n-2}, \cdots, a_0$). We call this phenomenon *Neighbor Based Importing and Exporting* (NBIE).

Because of the dynamic nature of the inter-domain routing system, signing every single path is worthless. We believe that even if a security schema only guarantees that *all authenticated paths are available path*, the protocol also can achieve *a similar level of security as S-BGP*. Inspired by the NBIE observation, we get rid of blindly signing every single path. NBIE abstracts the basic functionality of BGP. According to our measurement using the *whois* database, only a very small portion of routing polices violates the NBIE rule. In deed, our proposal can flexibly support complex routing polices, and we will discuss it in section 7.

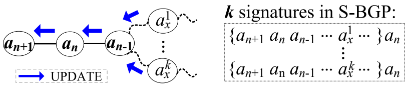

a_{n+1}
a_n
a_{n-1}
a_x^1
a_x^k
UPDATE
$\boldsymbol{k}$ signatures in S-BGP:
$\{a_{n+1}\ a_n\ a_{n-1} \cdots a_x^1 \cdots\}a_n$
$\{a_{n+1}\ a_n\ a_{n-1} \cdots a_x^k \cdots\}a_n$

Fig. 3. In S-BGP, a_n signs k paths which share a mutual path segment $\langle a_{n+1}, a_n, a_{n-1}\rangle$

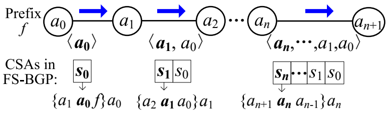

Prefix f
a_0
a_1
a_2
a_n
a_{n+1}
$\langle \boldsymbol{a_0} \rangle$
$\langle \boldsymbol{a_1}, a_0 \rangle$
$\langle \boldsymbol{a_n}, \cdots, a_1, a_0 \rangle$
CSAs in FS-BGP:
$\boldsymbol{s_0}$
$\boldsymbol{s_1}$ s_0
$\boldsymbol{s_n}$ $\cdots$ s_1 s_0
$\{a_1\ \boldsymbol{a_0} f\}a_0$
$\{a_2\ \boldsymbol{a_1}\ a_0\}a_1$
$\{a_{n+1}\ \boldsymbol{a_n}\ a_{n-1}\}a_n$

Fig. 4. Critical Segment Attestations (CSAs) in FS-BGP

4 FS-BGP: Fast Secure BGP

4.1 Overview

Following our key observation above, we propose Fast Secure BGP (FS-BGP) to secure AS paths. Given a available path p=$\langle a_{n+1}, a_n, ..., a_0 \rangle$, we define its set of *critical path segments* as $\{c_i, 0 \leq i \leq n\}$, where

$$c_i = \begin{cases} \langle a_1, a_0 \rangle & \text{for } i = 0 \\ \langle a_{i+1}, a_i, a_{i-1} \rangle & \text{for } 0 < i \leq n \end{cases}$$

and we call a_i the *owner* of c_i. Particularly, c_0 is called the *origin* critical path segment owned by a_0. A critical path segment $\langle a_{i+1}, a_i, a_{i-1} \rangle$ actually describes a routing export policy of its owner a_i, and implies that a_i can export all routes imported from a_{i-1} to a_{i+1}.

More specifically, FS-BGP uses Critical Segment Attestations (CSAs) to authenticate AS path. A CSA is simply the signature of the critical path segment signed by its owner. In a path p=$\langle a_{n+1}$, a_n, $\cdots$, $a_0 \rangle$, the CSA s_i signed by AS a_i is defined as:

$$s_i = \begin{cases} \{a_1\ a_0\ f\}a_0 & \text{for } i = 0 \\ \{a_{i+1}\ a_i\ a_{i-1}\}a_i & \text{for } 0 < i \leq n \end{cases}$$

The inclusion of the prefixes f in s_0 is necessary, because a_0 might be multihoming and only announces part of its prefixes to a_1.

Fig. 4 and Fig. 2 compare the signatures in FS-BGP and S-BGP, where FS-BGP and S-BGP recursively verify the path segments and path suffixes respectively. It is obvious that the number of distinct critical path segments is far less than the number of distinct paths. As a result, even using a small cache, the number of signing and verification operations in FS-BGP can be greatly reduced. In Fig. 3, a_n needs to sign each of the k paths individually in S-BGP. However, in FS-BGP, all the k different paths can reuse one signature of the common critical segment $\langle a_{n+1}$, a_n, $a_{n-1} \rangle$.

We argue that, under the NBIE rule, if *every* AS along a path signs the corresponding critical segment it owns, then the path can be authenticated as a available path. We will prove this claim in section 4.2. However, it is possible to forge an available but unannounced path if the security mechanism relies on CSA only, as shown in section 4.3. We will provide an effective solution, Suppressed Path Padding (SPP), to solve this problem in section 4.4.

4.2 Path Authentication in FS-BGP

In this section, we introduce our main argument on CSA based path authentication. We first define some notations as follows. We denote the set of available paths by $\mathcal{P}_A$, and the set of authenticated paths in S-BGP by $\mathcal{P}_S$. $\mathcal{P}_S$ exactly includes actually announced available paths.

We know $\mathcal{P}_S \subset \mathcal{P}_A$, i.e., S-BGP protects a subset of available paths. For FS-BGP, we define the set of all authenticated critical segments as $\mathcal{C}$, and use $\mathcal{P}_{FS}$

to represent the set of paths that can be authenticated using CSAs of $\mathcal{C}$. So $\mathcal{P}_{FS}$ is actually constructed by concatenating those path segments in $\mathcal{C}$. Generally, a constructed path $p_{i-1} = \langle a_i, a_{i-1}, \cdots, a_0 \rangle$ in $\mathcal{P}_{FS}$ must end with an origin critical segment in the form of $c_0 = \langle a_1, a_0 \rangle \in \mathcal{C}$, and can be extended to a longer path p_i by prepending *exactly one* AS a_{i+1} such that $\langle a_{i+1}, a_i, a_{i-1} \rangle \in \mathcal{C}$. Formally, we represent the concatenating procedure by an operator $\odot$ as

$$\langle a_{i+1}, \boldsymbol{a_i}, \boldsymbol{a_{i-1}} \rangle \odot \langle \boldsymbol{a_i}, \boldsymbol{a_{i-1}}, \cdots, a_0 \rangle = \langle a_{i+1}, a_i, a_{i-1}, \cdots, a_0 \rangle$$

We also take for granted that all paths considered here are loop-free, since BGP will drop such paths.

Theorem 1. *Under the NBIE assumption, we have $\mathcal{P}_S \subset \mathcal{P}_{FS} \subset \mathcal{P}_A$. That is, paths authenticated by S-BGP can also be authenticated by FS-BGP, and paths authenticated by FS-BGP are guaranteed to be available.*

Proof. The first part, $\mathcal{P}_S \subset \mathcal{P}_{FS}$, is straightforward. We will prove $\mathcal{P}_{FS} \subset \mathcal{P}_A$ by induction. Since the receiver of a path is always included in the signature, AS paths have lengths of at least two.

1) The case for a path $p = \langle a_1, a_0 \rangle \in \mathcal{P}_{FS}$ of length two is trivial.

2) Suppose all paths of length less than $k+1$ that are authenticated by FS-BGP are available. Given a path p_k of length $k+1$ such that $p_k = \langle a_{k+1}, a_k, a_{k-1}, \cdots, a_0 \rangle \in \mathcal{P}_{FS}$, p_k can only be constructed by a path $p_{k-1} = \langle a_k, a_{k-1}, \cdots, a_0 \rangle \in \mathcal{P}_{FS}$ of length k and a critical path segment $\langle a_{k+1}, a_k, a_{k-1} \rangle \in \mathcal{C}$. Due to the induction hypothesis, p_{k-1} is available, then p_k is also available.

By induction, any path that is authenticated by FS-BGP is guaranteed to be available. That is, $\mathcal{P}_{FS} \subset \mathcal{P}_A$.

The implication of Theorem 1 is that, ASes using FS-BGP can still implement the basic BGP functionality in a secure way ($\mathcal{P}_S \subset \mathcal{P}_{FS}$), and can not arbitrarily forge paths, since even if a path is forged, it is still available ($\mathcal{P}_{FS} \subset \mathcal{P}_A$).

4.3 Forged Paths in FS-BGP

This section analyzes the attack faced by FS-BGP, namely, an AS using FS-BGP can construct paths that are not actually announced by others, but avoid CSA based detection. According to Theorem 1, only paths in $\mathcal{P}_{FS} - \mathcal{P}_S$ can be forged and bypass the CSA based verification, and $\mathcal{P}_{FS} - \mathcal{P}_S \subset \mathcal{P}_A - \mathcal{P}_S$. That is, we only need to consider paths in $\mathcal{P}_A - \mathcal{P}_S$, which are sub-optimal and suppressed paths. Such paths can be constructed by concatenating critical segments.

In Fig. 5, p_a and p_b are two authenticated paths received by a_m, and they share a mutual path segment $\langle a_{i+1}, a_i \rangle$. Using critical segments in these two paths[5], a_m can construct a path p_d. According to Theorem 1, p_d is available and can pass the verification, but it may have never been announced before.

We use $p(a, :)$ and $p(:, a)$ to represent the suffix and prefix of the path p, starting from or ending with a respectively. As shown in Fig. 5, consider a

[5] Forging a path using critical segments in more than two paths is similar.

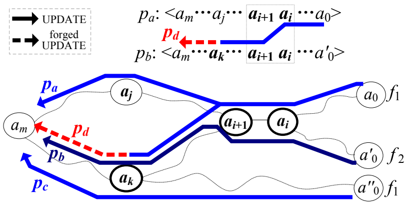


Fig. 5. Manipulator a_m forges path p_d

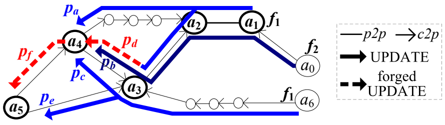


Fig. 6. a_4 hijacks a_5's traffic to f_1 by forging the path $p_f = \langle a_5, a_4, a_3 \rangle \odot p_d = \langle a_5, a_4, a_3, a_2, a_1 \rangle$

general case where an intermediate a_k receives two paths $p_d(a_k, :)$ and $p_c(a_k, :)$, both of which can reach prefix f_1. Although a_k ranks $p_d(a_k, :)$ lower than $p_c(a_k, :)$ and does not announce $p_d(a_k, :)$, a_m can still forge the path p_d and propagate it successfully.

Such a forged path could be used for prefix hijacking, as demonstrated in Fig. 6.[6] In this example, a_3 prefers its customer path $p_c(a_3, :)$ to its provider path $p_d(a_3, :)$, so it will not announce $p_d(a_3, :)$ to a_4. However, a_4 receives two paths p_a and p_b with a mutual path segment $\langle a_2, a_1 \rangle$, and it can forge the path p_d for prefix f_1 by concatenating $p_b(, : a_2)$ and $p_a(a_2, :)$. In normal circumstances, a_5 will choose the six-hop path p_e to reach prefix f_1, thus its traffic to f_1 should be forwarded to a_3. However, if a_4 announces the forged five-hop path $p_f = \langle a_5, a_4, a_3 \rangle \odot p_d$ to a_5, a_5 will prefer the shorter path p_f, and forward traffic to a_4 instead of to a_3. As a result, a_4 successfully pollutes the routing information of a_5, and *effectively hijacks* a_5's traffic to prefix f_1.

Although forging paths is possible in FS-BGP, there are still many restrictions on how paths can be forged. First, a path can only be forged by combining non-forged paths which share mutual segments. Second, some part of a forged path must be treated as sub-optimal or suppressed by some AS along the path. Third, forged paths are still available, and can only be used for the right prefixes. Last, forged paths can not be very short.

4.4 Prevent Effective Hijacking

Although there are limitations on forged paths, prefix hijacking is still possible. In this section, we discuss solutions to prevent prefix hijacking. We only concern *effective hijacking*, in a sense that, the recipient of a forged route indeed changes its forwarding path. That is, if when there is no forged route, AS a_m announces p_m but AS a_v does not choose p_m (or a path end with p_m) as its optimal path; and when a_m announces a forged path p_f, a_v changes its optimal path and chooses p_f (or a path end with p_f). In this case, a_v is effectively hijacked by a_m. When a_v receives the forged path constructed by a_m, a_v's decision process will be triggered, as shown in Fig. 1. The necessary condition of an effective hijacking is provided by the following theorem.

[6] In this paper, we use *p2p* indicates edge connects two peer ASes, while *c2p* edge is represented as an arrow from customer AS to provider AS.

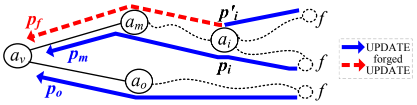


Fig. 7. Manipulator a_m announces a forged path p_f to its neighbor a_v

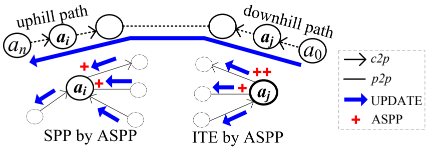


Fig. 8. ASPP for SPP and ITE

Theorem 2. *Under the NBIE assumption, if a forged path is no shorter than the non-forged path BGP should announce, it can not be used for effective hijacking.*

Proof. We first consider the direct recipient of a forged path. As shown in Fig. 7, the manipulator a_m forges a new path $p_f = \langle a_v, a_m, \cdots, p'_i\rangle$ when it wants to hijack traffic from a_v. Notice that p'_i is suppressed by a_i, while p_i is considered as optimal by a_i. The best route a_v originally chooses is $p_o = \langle a_v, a_o, \cdots\rangle$, where $a_o \neq a_m$ in an effective hijacking scenario.

Since both the LP and the rank of TB of p_m and p_f are the same,[7] and the Path Length of p_f is no shorter than that of p_m, p_f is no better than p_m in a_v's decision process. However, the original best route p_o is strictly better than p_m, so a_v will not prefer p_f to p_o. As a result, a_m can not effectively hijack prefix f from its neighbor a_v.

Actually, the analysis above does not depend on whether p_f is forged by a_m, or just part of it was previously forged by another AS and a_m just extends the forged path innocently, so our theorem holds for any circumstance.

We know that only suppressed path can be shorter than the optimal path. Thus, if there is a mechanism to guarantee that all suppressed paths are no shorter than their corresponding optimal paths, the manipulator can no longer effectively hijack a prefix either, according to Theorem 2. This idea can be implemented by using AS Path Pre-pending (ASPP). ASPP is a method to artificially increase the path length by padding multiple local ASNs in the front of an AS path [18]. We believe using ASPP to restrict the length of suppressed path will not bring additional burden to routers, since it is already widely used for Inbound Traffic Engineering (ITE).

We use the example in Figure 6 to explain how ASPP can be applied to FS-BGP. If a_3 intentionally increases the length of p_b by padding itself in the path, and only announces a route $p'_b = \langle a_4, a_3, a_3, a_3, a_2, a_1, a_0\rangle$, then a_4 can no longer forge a path short enough for effective hijacking. At the same time, when singing its critical segment $\langle a_4,\ a_3,\ a_2\rangle$, a_3 just needs to include the number of its occurrences in the corresponding CSA, i.e., $\{a_4, a_3, 3, a_2\}a_3$.

We call such a mechanism *Suppressed Path Padding* (SPP), and Algorithm 1 depicts the pseudo code for deciding how many times an AS a_i should pad itself. If a path is imported from a_{i-1} with the highest LP, a_i only appears once (line 1–2). Otherwise, k_i must be large enough such that no suppressed path can be shorter than the corresponding optimal path (line 4–7).

[7] The rank of TB may differ under some rare conditions. However, this is not critical for our theorem, and can be solved by just replacing "no shorter" with "longer".

Algorithm 1 Suppressed Path Padding

```
Input: local AS $a_i$, neighbor AS $a_{i-1}$
Output: $k_i$: number of times that $a_i$ needs to be added in the paths import from $a_{i-1}$
if $a_{i-1}$ has the highest $LP$ then
    return 1
$k_i \leftarrow 1$
for all path $p$ imported from $a_{i-1}$ do
    $opt(p) \leftarrow$ the optimal path corresponding to $p$
    if $PL(p) - PL(opt(p)) > k_i$ then
        $k_i \leftarrow PL(p) - PL(opt(p))$
return $k_i$
```

When local preferences are determined by business relationships, paths obey the valley-free rule: a path begins with zero or more *c2p* edges (*uphill path*), followed by zero or one *p2p* edge, and finally ends with zero or more *p2c* edges (*downhill path*) [4]. Fig. 8 compares SPP and ITE, both of which use ASPP. SPP only happens on the uphill path, as shown on the left side. Suppose a_i exports a route p_i which is *imported* from its provider (or peer), a_i need to pad itself in p_i. On the other side, ITE only happens along the downhill path when *exporting* routes to providers or peers. We can see that, SPP naturally expresses its own interests on neighbors, and has no side effect to ITE. We also note that, although using ASPP on suppressed paths for one prefix may affect routing for another prefix, it is still in the interest of the AS itself, since the AS already uses a lower preference to indicates its preference. As a conclusion, SPP is quite general, natural and easy to implement.

5 Security Level

Table 1 compares the security level of FS-BGP, S-BGP, and soBGP. Ineffective hijacks, such as false withdraw, selective dropping, or longer path announcing, can not effectively hijack a prefix. There are two types of attack, policy violating [5] and link-cut attack [3], which existing security schemes can not defend. As an autonomy organization, AS can completely determine its behaviour, so it is really hard to prevent it from violating its routing policy. For a link-cut attack, it is mainly achieved by wild destroying (i.e., DDoS attack). So defending against this kind of attack should be done through enhancing the robustness of BGP, since this paper aims to secure the AS path, we will not discuss link-cut attack.

We call a path is a *graph path* if it exist in the AS graph, and denote the set of graph paths by $\mathcal{P}_G$. We think that soBGP has a lower level of security compare with FS-BGP and S-BGP, since it can not defend against a cut-and-paste attack. In soBGP, attacker can easily forge an unavailable path through concatenating AS edges in the AS graph. However, we believe that FS-BGP can achieve a similar level of security as S-BGP.

Firstly, we divide all available paths $\mathcal{P}_A$ into four categories, as shown in Fig. 9: (1) outdated path, paths already announced but are temporary down; (2)

Table 1. Security Level Comparison for FS-BGP, S-BGP, and soBGP.

Type of Attack			**FS-BGP**	S-BGP	**FS-BGP (no SPP)**	soBGP
Ineffective hijack			✓	✓	✓	✓
False origin AS			✓	✓	✓	✓
Path not in the AS graph			✓	✓	✓	✓
$\mathcal{P}_G$	Unavailable path		✓	✓	✓	✘
	$\mathcal{P}_A$	Potential path	✓	✓	✓	✘
		Revealed path	✓*	✓	✘	✘
		Outdated path	✓*	✘	✘	✘
Policy violating [5]			✘	✘	✘	✘
Link-cut attack [3]			✘	✘	✘	✘

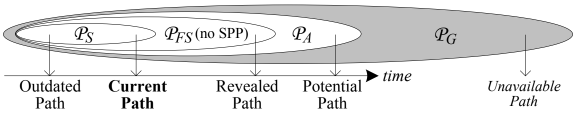


Fig. 9. Categories of available path.

current path, the recently announced path; (3) revealed path, paths constructed through concatenating authenticated critical segments; and (4) potential path, paths may be announced at some point in the future but can not be constructed even using received critical segments.

As failures occur in the global routing system, available paths are announced one after another. S-BGP actually authenticates outdated path and current path ($\mathcal{P}_S$). It can only use the expiration-date to roughly control the window which is exposed to outdated path replaying. Besides, the expiration-date must be long enough, otherwise there will be a UPDATE surge. According to Theorem 1, the light-weight version of FS-BGP (without SPP) can not defend against outdated path and revealed path attack, but it can defend against potential path attack. We claim that even without SPP, it is very difficult to launch an effective hijacking, since the average path length is very short and keeps on decreasing [20], and forged path can not be very short as it must be constructed by overlapped critical path segments. Armed with SPP, the full version of FS-BGP becomes more secure. It can defend against almost all revealed path and outdated path attack. This is because optimal path always has the longest live-time [11], and no path is shorter than optimal path after using SPP.

As a conclusion, we think that FS-BGP achieves a similar level of security as S-BGP.

6 Performance Evaluation

6.1 Methodology

We use real BGP UPDATE collected by RouterViews [1] to evaluate the performance. We consider S-BGP as the only mechanism that provides enough

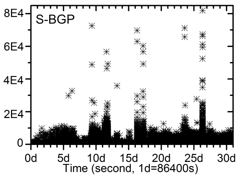


Fig. 10. INS (16K cache)

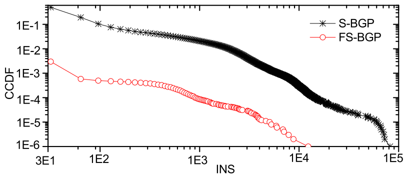


Fig. 11. CCDF of INS (16K cache)

security guarantee as FS-BGP, and compare their cost on signing and verification. Specifically, we use UPDATEs announced by the busiest monitor (a router in AS7018, owned by AT&T) and all UPDATEs received by the biggest collector (*route-views2*) during the whole month of August 2009 to evaluate how FS-BGP performs on a backbone router.

We assume ECDSA is used for signing and verification, as suggested by the IETF [17], and just count the number of signing/verification operations, since the cost of other operations is negligible compared to cryptographic operations. In the rest of this section, we consider two metrics as follows:

- *INS*: Instantaneous Number of Signings in each second.
- *INV*: Instantaneous Number of Verifications in each second.

Routers can use a cache to effectively improve the performance. Typically there is a limit on the cache size, then the number of cache misses measures the corresponding INS and INV.

6.2 Signing CSA

We use three different cache sizes for signing, i.e., 4K, 16K and 64K. If each signature occupies 256 bits [17], the memory cost will be around several megabytes, and is affordable for a backbone router. Since a signature always includes the corresponding recipient, one UPDATE message must be signed multiple times, once for each recipient. As a rough estimate, we use $m = 32$ as the average number of recipients for each UPDATE.[8]

Fig. 10 depicts the INS of FS-BGP and S-BGP in one month, under a moderate 16K cache size. In most of the time, the INS of FS-BGP is less than 100, while the INS of S-BGP often reaches up to tens of thousands. The maximum peak values of S-BGP reaches 81,920, while the maximum INS of FS-BGP is 12,365, only 15% of that of S-BGP. We also plot the Complementary Cumulative Distribution Function (CCDF) of INS in Fig. 11. In August 2009, there were UPDATE messages announced in 976,043 seconds. Only in 0.28% of that time does FS-BGP need to sign signatures, while the ratio in S-BGP is 44%.

6.3 Verifying CSA

Since there are much more signatures that need to be verified than to be signed, we use larger cache sizes (256K, 512K and 1024K entries) for comparing the

[8] As a scaling factor, the actual value of m is not important to our comparison.

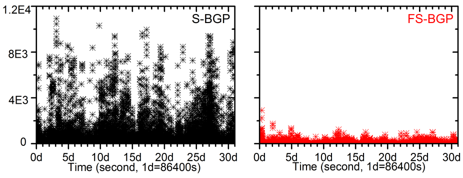


Fig. 12. INV (512K cache)

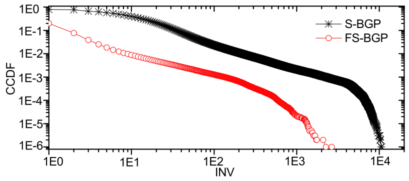


Fig. 13. CCDF of INV (512K cache)

Table 2. Performance Comparison for FS-BGP (FS) and S-BGP (S)

# Cache	Average INS		$\#(INS > 100)$		# Cache	Average INV		$\#(INV > 100)$	
Entries	FS	FS/S	FS	FS/S	Entries	FS	FS/S	FS	FS/S
4K	0.99	0.55%	507	0.48%	256K	2.35	8.8%	8121	11.3%
16K	0.50	0.56%	457	0.61%	512K	0.88	3.9%	3281	5.50%
64K	0.11	0.37%	133	0.54%	1024K	0.34	1.8%	1128	2.32%
∞(>4M)	0.08	1.70%	43	1.12%	∞(>13M)	0.34	6.7%	1128	11.7%

verification performance. FS-BGP has another advantage that once a critical segment is authenticated, only the path segment needs to be cached, but not the original signature.

Fig. 12 illustrates the INV of FS-BGP and S-BGP while using a moderate cache size of 512K. In most cases, the INV of S-BGP is ten times higher than that of FS-BGP, and sometimes even up to a hundred times. The maximum INV of FS-BGP is 2,919, only 26.7% of that of S-BGP, which is 10,921. The CCDF of INV in Fig. 13 shows that, only in 20% of the time when there are UPDATE messages does FS-BGP need to verify signatures, while S-BGP needs to verify signatures in 78% of the time.

Table 2 numerically compares the computational cost of FS-BGP and S-BGP. Using a cache of 16K entries, FS-BGP only needs to sign 0.56% as many messages as S-BGP. When using a moderate cache of 512K entries, the average verification cost of FS-BGP is only 3.9% of that of S-BGP. A large INS or INV (> 100) will delay the propagation of routing information, but it rarely happens in FS-BGP. In conclusion, FS-BGP performs orders of magnitude better than S-BGP, in both signing and verification. FS-BGP requires a very small cache, and can handle the most bursty BGP UPDATE messages, so it is an efficient and practical solution.

7 Discussions

Multiple Prefixes As noted before, when signing an origin critical segment, the corresponding prefix f should be included. In practice, an AS may own a large number of prefixes, and it is straightforward to extend the prefix f to a prefix set $\mathcal{F}$. Thus, $s_0 = \{a_1, a_0, k_0, \mathcal{F}\}a_0$, where k_0 is calculated by SPP, and $\mathcal{F}$ is the set of prefixes allowed to be announced to a_1. In practice, most of ASes announce all prefixes to their providers. Under these cases, FS-BGP can omit $\mathcal{F}$ in s_0 to represent no restriction on prefixes.

Complex Policies Our analysis till now are all based on the NIBE assumption in section 3. However, more complex policies also exist. The Routing Policy Specification Language (RPSL) [2] is commonly used by ASes. There are three kinds of transitive route filters in RPSL: prefix filters, AS path filters, and origin AS filters. We queried all ASes by *whois* and collected 758K import/export expressions in total. Among all these expressions, 1.1% of them use prefix filters. To support prefix filters, FS-BGP can sign the available prefixes together with the critical segment. AS path filters occur in 0.3% of policy expressions, and to support them, an AS can sign the full AS path. Since there is only a very small portion, the influence on computational cost is negligible. About 60% of policy expressions use origin AS filter. To support them, CSAs in FS-BGP can be extended to include the available origin ASes. In most cases, the number of available origin ASes is very small.

Nevertheless, the main purpose of route filters is to protect the routing system against distribution of inaccurate routing information [2]. The use of route filters is mainly due to security considerations rather than policy requirements. We believe that under a security framework (such as FS-BGP), these filters are not needed any more. Even if they do exist, FS-BGP can support them flexibly.

Privacy Concerns Internet is a commercialized network, an AS may not want to reveal its proprietary information (i.e., customer list) to its competitors. FS-BGP does not require an AS to disclose its proprietary information, since the critical segments are nothing new but already included in BGP UPDATE. FS-BGP does not allow others to obtain the information more easily either, by employing the existing distribution mechanism of BGP. No centralized or public database such as in IRV, soBGP or IRR need to be maintained. In conclusion, we believe FS-BGP preserves the privacy of a business entity.

8 Conclusion and Future Works

This paper introduces an efficient approach, FS-BGP (Fast Secure BGP), to secure AS path and prevent prefix hijacking. Through signing critical AS path segments, FS-BGP guarantees the authentication of all available paths. Through padding suppressed path, FS-BGP prevents almost all replay attacks. We prove that FS-BGP can achieve a similar level of security as S-BGP. In our evaluations based on BGP UPDATE data collected from real backbone routers, FS-BGP performs orders of magnitude better than S-BGP. By using even a very small cache, the signing and verification overhead of FS-BGP account for only 0.56% and 3.9% of that of S-BGP respectively. Indeed, signing and verification can always be accomplished as fast as the most bursty BGP UPDATE arrivals, which implies that FS-BGP will hardly delay the propagation of routing information. In addition, FS-BGP can flexibly support complex routing polices, and can preserve the privacy of an AS.

We plan to design more efficient cache replacement algorithm, and evaluate the influence on convergence time after deploying FS-BGP on a large scale.

Besides, we will also investigate the potential to use available paths constructed by critical path segments as backup paths in route protection.

References

1. The routeviews project. http://www.routeviews.org (2009)
2. Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D., Terpstra, M.: RFC 2622, routing policy specification language (RPSL). http://tools.ietf.org/html/rfc2622 (1999)
3. Bellovin, S.M., Gansner, E.R.: Using link cuts to attack Internet routing. http://hdl.handle.net/10022/AC:P:9052 (2003)
4. Gao, L., Rexford, J.: Stable Internet routing without global coordination. IEEE/ACM Trans. Netw. 9(6), 681–692 (2001)
5. Goldberg, S., Schapira, M., Hummon, P., Rexford, J.: How secure are secure interdomain routing protocols? In: SIGCOMM (2010)
6. Goodell, G., Aiello, W., Griffin, T., Ioannidis, J., McDaniel, P.D., Rubin, A.D.: Working around BGP: An incremental approach to improving security and accuracy in interdomain routing. In: NDSS (2003)
7. Hu, Y.C., Perrig, A., Sirbu, M.A.: SPV: secure path vector routing for securing BGP. In: SIGCOMM. pp. 179–192 (2004)
8. Karlin, J., Forrest, S., Rexford, J.: Pretty good BGP: Improving BGP by cautiously adopting routes. In: ICNP. pp. 290–299 (2006)
9. Kent, S., Lynn, C., Mikkelson, J., Seo, K.: Secure border gateway protocol (S-BGP). IEEE Journal on Selected Areas in Communications 18, 103–116 (2000)
10. Nicol, D.M., Smith, S.W., Zhao, M.: Evaluation of efficient security for BGP route announcements using parallel simulation. Simulation Modelling Practice and Theory 12(3-4), 187–216 (2004)
11. Oliveira, R., Zhang, B., Pei, D., Izhak-Ratzin, R., Zhang, L.: Quantifying path exploration in the Internet. In: Proc. of the 6th ACM SIGCOMM Internet Measurement Conference (IMC). Rio de Janeriro, Brazil (2006)
12. van Oorschot, P.C., Wan, T., Kranakis, E.: On interdomain routing security and pretty secure BGP (psBGP). ACM Trans. Inf. Syst. Secur. 10(3) (2007)
13. Rekhter, Y., Li, T., Hares, S.: RFC 4271: Border gateway protocol 4. http://tools.ietf.org/html/rfc4271 (2006)
14. RIPE: Youtube hijacking: A ripe ncc ris case study. http://www.ripe.net/news/study-youtube-hijacking.html (2008)
15. RIPE NCC: Resource certification. http://ripe.net/certification/ (2011)
16. Subramanian, L., Roth, V., Stoica, I., Shenker, S., Katz, R.H.: Listen and whisper: Security mechanisms for BGP. In: NSDI. pp. 127–140 (2004)
17. Turner, S.: BGP algorithms, key formats, & signature formats. http://tools.ietf.org/html/draft-ietf-sidr-bgpsec-algs (2011)
18. Wang, J.H., Chiu, D.M., Lui, J.C.S., Chang, R.K.C.: Inter-as inbound traffic engineering via ASPP. Transactions On Network And Service Management 3(1) (2007)
19. White, R.: Architecture and deployment considerations for secure origin BGP. http://tools.ietf.org/html/draft-white-sobgp-architecture (2006)
20. Xiang, Y., Yin, X., Wang, Z., Wu, J.: Internet flattening: Monitoring and analysis of inter-domain routing. In: IEEE ICC (2011)
21. Zmijewski, E.: Threats to internet routing and global connectivity. http://www.renesys.com/tech/presentations/pdf/20thAnnualFIRST.pdf (2008)