# Balanced Peer Lists: Towards a Collusion-Resistant BGP

Yan Li
Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712
Email: yanli@cs.utexas.edu

Mohamed G. Gouda
Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712
Email: gouda@cs.utexas.edu

***Abstract*—In BGP, Autonomous Systems (ASes) advertise routes. Unfortunately, malicious ASes can advertise false routes that do not exist in the Internet. Many extensions of BGP have been proposed to allow each AS to check whether the received routes are false. It turns out that none of these extensions can defend against collusions among malicious ASes. In this paper, we present an extension of BGP that can defend against collusions. In our extension, each listed AS in an advertised route supplies a certified full list of all its peers, i.e. neighbors. Because full peer lists can be very large, we develop an optimization where each AS in an advertised route supplies a balanced peer list that is much smaller than its full peer list. Using real Internet topology data, we demonstrate that the average, or largest, balanced peer list is 92% smaller than the average, or largest respectively, full peer list.**

## I. Introduction

The Internet is composed of a large number of connected Autonomous Systems, or ASes for short. Each AS in the Internet owns one or more IP prefixes and the ASes use the standard Border Gateway Protocol (BGP) [18] to advertise routes to every IP prefix in the Internet. Each advertised route lists all the ASes on the route. In BGP, a malicious or misconfigured AS can advertise to its peers false routes that do not correspond to existing routes in the Internet. When this happens, none of the peers can detect that the advertised routes are indeed false, and they may proceed to adopt these false routes and use them to route packets [9]. This vulnerability of BGP is usually referred to as IP prefix hijacking [30].

When packets are routed along false routes, three erroneous outcomes can arise [30]: packet discarding which can yield a DoS attack; packet sniffing which can cause privacy leakage [5]; and destination impersonation which can be used to create false identities in spamming [17]. The Internet has already suffered some notable instances of IP prefix hijacking that resulted from the misconfiguration of some ASes [1], [3], [14], [16]. In one instance [16], packet flows to more than 10,000 ASes were discarded.

Many techniques have been recently proposed to address IP prefix hijacking in BGP. These techniques can be classified into two categories: prevention techniques [4], [7], [9], [10], [21]–[23], [27] and detection techniques [6], [8], [11], [12], [19], [26], [29], [30]. In a prevention technique, BGP is extended such that when an AS receives an advertised route from any one of its peers, the receiving AS can immediately check whether the advertised route is false and should be discarded right away. In a detection technique, BGP remains unchanged, but an additional protocol is developed to monitor the adopted routes in all the ASes and detect whether any of them is false and should be discarded.

Clearly prevention techniques are more responsive and more vigilant against IP prefix hijacking than detection techniques. Thus, a prevention technique needs to be employed to extend BGP and make it secure against the hijacking of IP prefixes. Unfortunately, none of the prevention techniques that have been proposed so far can prevent the hijacking of IP prefixes caused by a "collusion" of several malicious or misconfigured ASes. Several (malicious or misconfigured) ASes in the Internet constitute a collusion if these ASes share among themselves all the cryptographic keys that each of them owns and they agree to act in concert to hijack IP prefixes in BGP. In this paper, we propose the first ever technique to prevent the hijackings of IP prefixes even those which are caused by collusions of ASes.

Next we illustrate how a collusion can succeed in hijacking IP prefixes against a good prevention technique, similar to S-BGP [9], [10]. Assume that BGP is extended, to prevent IP prefix hijacking, as follows. Each AS is supplied with both a private key and a public key. An AS is supposed to know only its own private key and the public key of every other AS in the Internet. But for ASes in a collusion, it is possible that each AS in a collusion may know the private keys of all ASes in the collusion. When an AS receives a route to some IP prefix and wishes to adopt it, the AS adds itself to the route and signs the augmented route using its private key before forwarding it to every one of its peers. Now assume that AS x receives a route to an IP prefix P, and assume that the received route goes from AS u (which owns P), to AS v, to AS w, before it is received by AS x. This route has already been signed by the private keys of u, v, and w. But in this case, if both v and x are in a collusion and x knows the private key of v, then x can shorten the route and make it consist of only two ASes u and v, before x adopts and advertises the shorter route. Note that by shortening the route, x creates a false route but, in doing so, it makes the shorter route more attractive for adoption by other ASes.

978-1-4244-4581-3/09/$25.00 ©2009 IEEE

The basic idea of our technique to prevent the hijackings of IP prefixes, even those that are caused by collusions of ASes, is quite simple. When an AS receives a route and wishes to adopt it, the AS adds itself to the route, then adds a signed "full list" of all its peers to the route specification, then it advertises the augmented route to every one of its peers. Each full peer list in an advertised route is signed by the private key of a trusted central authority whose public key is known to all the ASes in the Internet. When an AS receives an advertised route, it first checks that each signed full peer list on the route is correct, then it uses these full peer lists to check that the route is true. Note that no individual AS and no collusion of ASes can modify or forge any of the full peer lists on an advertised route.

The only problem of our technique is that the full peer lists of some ASes in the Internet are very large containing thousands of peers. Even worse these ASes appear in most advertised routes. Therefore the specifications of most advertised routes will become large. To solve this problem, we show that the signed full peer lists in an advertised route can be replaced by signed balanced peer lists which are much smaller in size. The full version of this paper can be found in [13].

## II. A Model of BGP and its Attacks

In this paper, we abstract the AS-level topology of the Internet by an undirected graph G, called the *Internet topology graph*. In G, each node is called an *Autonomous System*, or *AS* for short, and each undirected edge $\{u, v\}$ between two ASes u and v is called a *peering* between u and v. If G has a peering $\{u, v\}$, then u is called a *peer* of v and v is called a *peer* of u.

Each AS in the Internet topology graph G *owns* a set of IP prefixes. The ASes in G exchange BGP update messages, each of which is of the following format:

$$(P : (u,..,v))$$

where P is an IP prefix and (u,..,v) is a sequence of ASes that represents a simple path in G, and so each pair of adjacent ASes in the sequence is a peering in G. The first AS in the sequence, namely AS u, is the *owner* of prefix P.

Consider the case where an AS u in the Internet topology graph G owns an IP prefix P. In this case, u initially sends a copy of the BGP update message (P : (u)) to each one of its own peers. When a peer v of u receives the BGP update message (P : (u)) and decides based on its routing policy that the best route to reach the IP prefix P is to go from v to u, then v sends a copy of the BGP update message (P : (u,v)) to each one of its own peers other than u, and the cycle repeats.

When an AS w receives a BGP update message (P : (u,..,v)), it makes the following two assumptions. First, w assumes that the sequence (u,..,v) in the received message constitutes a simple path in the Internet topology graph G. Second, w also assumes that u is the owner of the IP prefix P. Unfortunately, these two assumptions may be wrong since malicious or misconfigured ASes could have corrupted the contents of the update message. Consequently, BGP is vulnerable to two types of attacks: invalid owner attacks and invalid route attacks.

In an *invalid owner attack*, a malicious or misconfigured AS v sends a copy of the BGP update message (P : (u,..,v)) to each one of its peers even though u is not the owner of the IP prefix P. To overcome this type of attack, whenever an AS w receives a BGP update message (P : (u,..,v)), w should be able to check whether u is the true owner of prefix P. This can be accomplished by attaching, to each update message (P : (u,..,v)), a signed certificate certifying that the owner of prefix P is AS u. This certificate is to be signed by the private key of a trusted central authority.

In an *invalid route attack*, a malicious or misconfigured AS v sends a copy of the BGP update message (P : (u,..,v)) to each one of its peers even though the sequence of ASes (u,..,v) does not constitute a simple path in the Internet topology graph G. This means that at least two adjacent ASes in the sequence (u,..,v) are not peers in G. Therefore, to overcome this type of attack, whenever an AS w receives a BGP update message (P : (u,..,v)), w should be able to check whether each two adjacent ASes in the sequence (u,..,v) are peers in G.

In Sections III through V, we present techniques to overcome invalid route attacks.

## III. Route Validation Using Full Peer lists

In this section we describe a technique that can be used to check the validity of the AS sequences in BGP update messages while defend against collusion. This technique is based on three assumptions.

- ***Central Authority***: There is an Internet central authority, which knows the list of all the peers for each AS in the Internet topology graph G. Let FPL.u denote the full list of all peers for AS u.
- ***Public and Private Keys***: This central authority has a pair of public and private keys. Moreover, every AS in the Internet topology graph G knows the public key of this central authority.
- ***Peer Certificates***: This central authority generates a peer certificate for each AS in the Internet topology graph G and gives a copy of the peer certificate to the AS. The peer certificate, generated by the central authority, for an AS u is defined as

  $$PC.u = (u : FPL.u : signature)$$

  where FPL.u is the full list of all peers of AS u, and "signature" is the digital signature of (u: FPL.u) using the private key of the central authority. When an AS v receives the peer certificate PC.u, AS v can use its copy of the public key of the central authority to check that PC.u has not been tampered with after it was generated by the central authority.

(Note that the assumption of a central authority that issues certificates to the ASes in the Internet has been proposed earlier in order to validate the true owners of IP prefixes [9], [10]. Therefore, our three assumptions merely expand the role

of this central authority from simply issuing owner certificates to issuing both owner and peer certificates.)

Based on the above three assumptions, we propose to redefine a BGP update message to become as follows.

$$(P: (u,..,v) : OC.P : (PC.u,..,PC.v))$$

where OC.P is the owner certificate of the IP prefix P. It is defined as follows.

$$OC.P = (P : u : signature)$$

where u is the AS that owns prefix P, and "signature" is the digital signature of (P : u) using the private key of the central authority.

In general, when an AS w receives a BGP update message (P: (u,..,v) : OC.P : (PC.u,.., PC.v)) from a peer v, w needs to check that the sequence (u,..,v) constitutes a simple path in the Internet topology graph G. Thus, w needs to check that each pair of adjacent ASes in the sequence (u,..,v) is a peering in G. To check whether a pair {x,y} of adjacent ASes in (u,..,v) is a peering in G, AS w needs to check whether x is an element in FPL.y which is part of PC.y in the received BGP update message. AS w concludes that {x,y} is a peering in G iff x is in FPL.y.

We refer to this technique for validating the routes in BGP update messages as the *full peer list technique*.

The maximum degree dmax of an AS in the Internet topology graph G is large and it is getting larger over time. For example, as discussed below, the value of dmax in January, 2007 is more than 2800. Thus, the complexity of using full peer lists to check the validity of routes in BGP update messages is unacceptably high. In the next section, we propose to use balanced peer lists, which are often much smaller than full peer lists, to check the validity of routes in BGP update messages.

## IV. Route Validation Using Balanced Peer Lists

We assume that the Internet central authority imposes a hierarchy of levels on the Internet topology graph G such that each AS in G is assigned to exactly one level in the imposed hierarchy.

If an AS u in G is assigned to level i, where i is in the range of 0..L-1 and L is the number of levels in the imposed hierarchy on G, then the level of u, denoted L.u, is i.

The *balanced peer list* for an AS u, denoted BPL.u, is defined as the list containing every peer v (of u) that satisfies one of the following two conditions:

*i.* the level of v is higher than the level of u.
*ii.* the level of v equals the level of u and the identifier of v is larger than the identifier of u.

If the balanced peer list BPL.u of an AS u has an AS v, then v is called a *balanced peer* of u. Note that the balanced peer relationship is asymmetric. In other words, if v is a balanced peer of u, then u is not a balanced peer of v.

Given the balanced peer lists BPL.u and BPL.v of two ASes u and v, one can determine whether or not the pair {u,v} is a peering in the Internet topology graph G as follows. If either u is in BPL.v or v is in BPL.u, then {u,v} is a peering in G, otherwise {u,v} is not a peering in G.

The number of balanced peers of an AS u, or the size of the balanced peer list BPL.u, is called the *balanced degree* of u. The maximum balanced degree of an AS in the Internet topology graph G after imposing a hierarchy on it is denoted *bmax*.

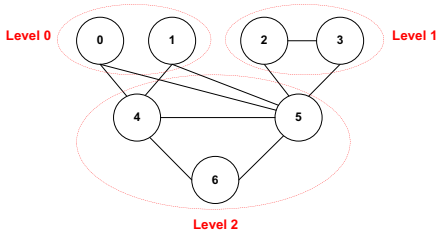


Fig. 1. An example of a 3-level hierarchy imposed on a graph G

As an example, consider the simple graph G in Figure 1. This graph has seven ASes whose identifiers are in the range 0..6. Assume that a hierarchy of three levels, level 0 through level 2, is imposed on this graph. ASes 0 and 1 are assigned to level 0 in the hierarchy. ASes 2 and 3 are assigned to level 1 and ASes 4, 5, 6 are assigned to level 2. The balanced peer list for each AS in this graph is as follows:

BPL.0 = {4, 5}
BPL.1 = {4, 5}
BPL.2 = {3, 5}
BPL.3 = {5}
BPL.4 = {5, 6}
BPL.5 = {6}
BPL.6 = {}

Therefore, the maximum balanced degree bmax of this graph G is two, which is much smaller than the maximum degree dmax of G which is six.

It follows that it is beneficial to redefine the peer certificate PC.u, generated by the central authority for each AS u, as follows.

$$PC.u = (u : BPL.u : signature)$$

Now when an AS w receives a BGP update message (P: (u,..v) : OC.P : (PC.u,..PC.v)) from a peer v, w checks whether each pair {x,y} of adjacent ASes in the sequence (u,..,v) is a peering in the Internet topology graph G using BPL.x (which occurs in the certificate PC.x) and BPL.y (which occurs in the certificate PC.y).

We refer to this technique for validating the routes in the BGP update messages as the *balanced peer list technique*.

In the next section, we identify a class of hierarchies that can be imposed on the Internet topology graph G and can yield bmax'es that are much smaller than the dmax of G.

## V. Route Validation Using Degree-Based Hierarchies

A *degree-based hierarchy* of L levels that can be imposed on the Internet topology graph G is defined by a sequence of L+1 integers

$$(d[0], d[1], .., d[L])$$

where 0 = d[0] < d[1] < .. < d[L] = dmax and dmax is the maximum degree of an AS in G. An AS u is said to be *in level* i of this degree-based hierarchy, where i is in the range 0..(L-1), iff the degree of u is in the interval (d[i], d[i+1]].

A degree-based hierarchy (d[0], d[1], .., d[L]) that is imposed on the Internet topology graph G is called *R-bounded*, for some integer R, iff for each AS u, which is in some level i of the imposed hierarchy on G, u has at most R peers that are in level i or higher in the imposed hierarchy.

Note that this definition of R-boundedness holds trivially if R is at least dmax. Therefore, it is our intention to use this definition only when R is much smaller than dmax.

Next, we describe an algorithm, Algorithm 1 below, that takes as inputs an Internet topology graph G and a small integer R and computes a degree-based hierarchy (d[0], d[1], .., d[L]) of L levels that can be imposed on G.

**Algorithm 1**: Compute a degree-based hierarchy that can be imposed on a given Internet topology graph G

```
   input : the degree distribution function F(e) of G,
           a small integer R
   output: a degree-based hierarchy (d[0], .., d[L]) that can be
           imposed on G
 1 begin
 2     d[0] := 0;
 3     d[1] := R;
 4     i := 1;
 5     while d[i] < dmax do
 6         i := i + 1;
 7         d[i] := R / (1 - F(d[i-1]));
 8         while d[i] == d[i-1] & d[i] < dmax do
 9             d[i-1] := d[i-1] + 1;
10             d[i] := R / (1 - F(d[i-1]));
11         end
12     end
13     L := i;
14     d[L] := dmax;
15 end
```

In fact, instead of taking the actual Internet topology graph G as an input, Algorithm 1 takes an abstraction of G called the *degree distribution function* F(e) of G. The value of F(e) is the percentage of ASes in G whose degrees are at most e. Thus, the value of F(e) is defined only when e is in the range 1..dmax where dmax is the maximum degree of an AS in G.

After using Algorithm 1 to compute a degree-based hierarchy (d[0], .., d[L]) that can be imposed on the Internet topology graph G, the resulting balanced peer list for every AS in G can be computed by Algorithm 2 below. Algorithm 2 takes as inputs G and (d[0], ..,d[L]) and computes BPL.u for every AS u in G. Algorithm 2 has two phases. In the first phase, the algorithm computes the level L.u for every AS u in G, and in the second phase, it uses the computed levels to compute BPL.u for every AS u in G.

## VI. Experiments

In this section we evaluate the results of applying Algorithms 1 and 2 to the real Internet topology graph of January 1, 2007 [15], [25] and show that these two algorithms can still achieve dramatic reduction in the size of the peer list for every AS in the Internet.

**Algorithm 2**: Compute the balanced peer list for every AS in an Internet topology graph G upon which a given degree-based hierarchy (d[0], .., d[L]) is imposed

```
   input : an Internet topology graph G,
           a degree-based hierarchy (d[0], .., d[L])
           that is imposed upon G
   output: a balanced peer list BPL.u for each AS u in G
 1 begin
 2     for each AS u in G do
 3         find i where degree of u is in the
 4         interval (d[i-1], d[i]];
 5         L.u := i;
 6     end
 7     for each AS u in G do
 8         for each peer v of u do
 9             if (L.v > L.u) or (L.v = L.u & v > u) then
10                 add v to BPL.u;
11             end
12         end
13     end
14 end
```

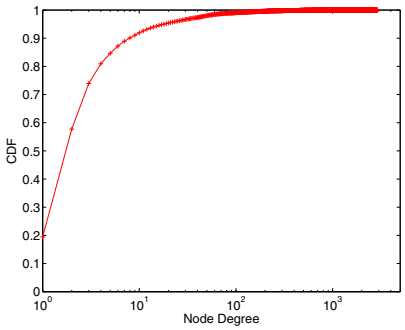


Fig. 2. Distribution of the AS degrees in the Internet of Jan. 1, 2007

The degree distribution for the Internet topology graph of January 1, 2007, is shown in Figure 2. There were roughly 25000 ASes in the Internet. The degree distribution of the Internet topology graph follows power-law distribution [20]. Because of this power law distribution, a few levels are needed to impose a degree-based hierarchy on the Internet topology graph according to the computation of d[i], $0 \leq i \leq L$. The degree-based hierarchy for the Internet topology graph, computed by applying Algorithm 1 with R = 10, is shown in the first three columns in Table I.

Note that based on the characteristics of the power-law degree distribution, the number of ASes with very high degree is quite small. Thus when applying Algorithm 1 to impose a degree-based hierarchy on the Internet topology graph, the number of ASes in the last level L-1 may be smaller than R. This means that some high-degree ASes which can be put in the last level end up in level L-2. The more high-degree ASes in a level, the higher the maximum balanced degree is.

TABLE I
COMPARISON BETWEEN BPLS AND FPLS AFTER IMPOSING A DEGREE-BASED HIERARCHY WITH R = 10 UPON THE INTERNET OF JAN. 1, 2007

| Level i | d[i] | #AS | max \|FPL\| | max \|BPL\| | avg \|FPL\| | avg \|BPL\| |
|---|---|---|---|---|---|---|
| 0 | 10 | 22641 | 10 | 10 | 3.8 | 3.4 |
| 1 | 124 | 1800 | 124 | 107 | 48.6 | 24.9 |
| 2 | 791 | 178 | 791 | 121 | 325.3 | 42.3 |
| 3 | 2886 | 11 | 2886 | 10 | 1780.9 | 5.0 |
| all G | - | 24630 | 2886 | 121 | 262.2 | 20.2 |

So in order to make the maximum balanced degree of ASes in level L-2 as small as possible, we introduce a simple but effective optimization when applying Algorithm 1: when the last level ends up with less than R+1 ASes (because the high-degree ASes probably are connected to each other to form a clique, the balanced degree of those ASes will be R if there are R+1 ASes in that level), we will make it R+1 and change the corresponding maximum degree d[L-1] for level L-2.

Next, we evaluate the maximum and average degree (size of FPLs) and balanced degree (size of BPLs) for each level and the whole graph G. The difference of the maximum balanced degree of G from R is the deviation of the output hierarchy from R-boundedness. Since ASes with higher degrees (transit ASes or tier-1 ASes) usually appear more often in the AS sequences in BGP update messages than ASes with lower degrees (stub ASes), when we calculate the average degree (or balanced degree), the degree (or balanced degree) of each AS is counted proportional to its degree. For example, the average balanced degree of G is calculated as $\sum_{\forall AS\ u} |BPL.u| \times \frac{|FPL.u|}{\sum_{\forall AS\ v} |FPL.v|}$, where $|FPL.u|$ (or $|BPL.u|$) denotes the size of the full (or balanced) peer list of u. Also, we calculate the average degree and balanced degree for each level in the same way.

The maximum and average sizes of BPLs for the degree-based hierarchy when R = 10, compared to those of FPLs are shown in the right four columns in Table I. Since the actual degrees of nodes in the lower two levels, Level 0 and 1, are relatively small, the reduction in the maximum and average sizes of peer lists using BPLs compared to that using FPLs for these two levels is not remarkable (14% in maximum and 49% in average for level 1). However, both the maximum and average sizes of peer lists of ASes in higher levels are greatly reduced. For level 2, the maximum degree achieves 85% reduction. The same trend happens for the average size, which is reduced by 87% if using BPLs. The reduction in the last level is even prominent, 99.7% reduction in both maximum size and average size. Finally the last row shows that using the BPL technique, we achieve over 92% reduction in both the maximum and average sizes of peer lists for all the ASes in the Internet topology graph (24630 ASes totally). With input R = 10, we end up with 121-bounded hierarchy instead of 10-bounded hierarchy.

Now, We investigate the effect of input R on the degree-based hierarchy. When we lower the input R to 2, the resulting degree-based hierarchy from Algorithm 1 is shown in the left three columns in Table II.

The maximum and average sizes of BPLs for degree-based hierarchy with R = 2, compared to those of FPLs, are shown in the right four columns in Table II. By using a smaller R, we end up with more levels in the hierarchy but with a lower bmax (103 compared to 121).

TABLE II
COMPARISON BETWEEN BPLS AND FPLS AFTER IMPOSING A DEGREE-BASED HIERARCHY WITH R = 2 UPON THE INTERNET OF JAN. 1, 2007

| Level i | d[i] | #AS | max \|FPL\| | max \|BPL\| | avg \|FPL\| | avg \|BPL\| |
|---|---|---|---|---|---|---|
| 0 | 2 | 14213 | 2 | 2 | 1.80 | 1.78 |
| 1 | 5 | 6636 | 5 | 5 | 3.69 | 3.36 |
| 2 | 14 | 2298 | 14 | 14 | 9.2 | 6.2 |
| 3 | 34 | 734 | 34 | 34 | 23.6 | 12.1 |
| 4 | 66 | 405 | 66 | 61 | 49.5 | 26.6 |
| 5 | 143 | 189 | 143 | 103 | 103.2 | 38.3 |
| 6 | 318 | 102 | 313 | 78 | 215.0 | 38.9 |
| 7 | 791 | 42 | 668 | 40 | 489.7 | 20.2 |
| 8 | 2886 | 11 | 2886 | 10 | 1780.9 | 5.0 |
| All G | - | 24630 | 2886 | 103 | 262.2 | 16.7 |

## VII. RELATED WORK

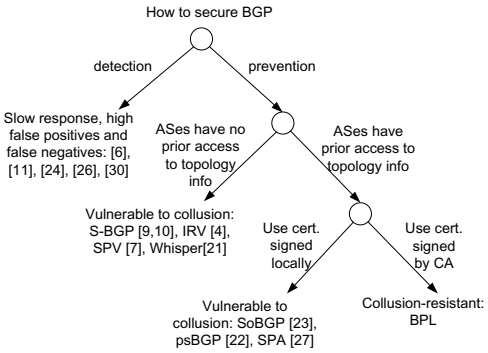


Fig. 3. Classification of techniques proposed to secure both owner and route in BGP

Figure 3 shows a classification of proposed techniques that can defend against both invalid owner and invalid route attacks.

S-BGP [9], [10] uses *Route attestations* signed by each AS in the sequence of the update message, to allow the next hop AS to propagate the route. It provides strong route validation only when there is no collusion, and it also suffers from high computation and storage overhead.

Subsequent work [4], [7], [22], [27] only focuses on optimizations to reduce the computation or space overhead of S-BGP, but all of them is vulnerable to collusion. The whisper model in [21] detects inconsistency of two advertisements using cryptographic functions. But it can not guarantee deterministic detection. Moreover, it can not defend against collusion either.

In SoBGP [23], each AS keeps a database of network topology by exchanging a full list of its own peers. The maintained topology is then used to check the route plausibility. However, the peering information can be quite large and thus incur both bandwidth and storage overhead. Furthermore, since the certificates are signed by locally by ASes themselves, it can not handle collusion, either.

Since [2] only focuses on authenticating the owner, it is not included in the branch of prevention techniques in Figure 3.

Some other work has been done based on various anomaly detection techniques to detect whether there is a hijack [6], [8], [11], [12], [19], [24], [26], [28]–[30]. Although most of these anomaly detection techniques can be deployed incrementally since no changes to BGP are needed, they either have high false positives and false negatives or depend on careful selection of probing locations to improve detection accuracy. Moreover, in general anomaly detection requires quick response of administrators to analyze the cause of the problem and to eliminate false positives or false negatives, which is hard to guarantee in practice.

## VIII. Concluding Remarks

In this paper we have presented the first ever technique to prevent the hijackings of IP prefixes in BGP when these hijackings are attempted by any collusion of ASes in the Internet. The technique is based on the idea of attaching to each advertised route in BGP a sequence of peer certificates, one certificate for each listed AS in the advertised route. We showed that each peer certificate of an AS u can include a balanced peer list of u, instead of the full peer list of u, since a balanced peer list of u is usually smaller than the full peer list of u.

We also presented two algorithms. The first algorithm can be used to impose a (degree-based) hierarchy on the Internet topology graph. And the second algorithm can be used to utilize the imposed hierarchy to compute a "small" balanced peer list for each AS in the Internet. We have applied these two algorithms on the Internet topology graph of January 1, 2007 and showed that the maximum (or average, respectively) size of a BPL of an AS in the Internet is 92% smaller than the maximum (or average, respectively) size of a FPL of an AS. We have achieved similar results when we applied our two algorithms on the Internet topology graph of other dates.

In our proposed technique, the peer certificates, and the owner certificates for that matter, are signed by a central authority. This design decision is critical to the correctness of our technique. For instance, if the peer certificate of an AS is allowed to be signed by a private key that belongs to the AS, then the signed peer certificate can be wrong since ASes can lie and can collude due to maliciousness or misconfiguration.

In our technique, the update messages of BGP need to be expanded to include owner and peer certificates. But It is possible to consider keeping the update messages unchanged and creating a distributed database, similar to DNS, that hosts all owner and peer certificates that are signed by the central authority.

## References


[1] "Nanog mailing list," http://www.nanog.org/mailinglist.html.

[2] W. Aiello, J. Ioannidis, and P. McDaniel, "Origin authentication in interdomain routing," in *CCS*, 2003.

[3] V. J. Bono, "7007 explanation and apology," http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html, April 1997.

[4] G. Goodell, W. Aiello, T. Griffin, J. I. nad P. McDaniel, and A. Rubin, "Working around BGP: An incremental approach to improving security and accuracy in interdomain routing," in *NDSS*, 2003.

[5] X. Z. Hitesh Ballani, Paul Francis, "A study of prefix hijacking and interception in the internet," in *Sigcomm*, 2007.

[6] X. Hu and Z. M. Mao, "Accurate real-time identification of ip prefix hijacking," in *IEEE Symposium on Security and Privacy*, 2007.

[7] Y. Hu, A. Perrig, and M. Sirbu, "SPV: Secure path vector routing for securing BGP," in *SIGCOMM*, 2004.

[8] J. Karlin, S. Forrest, and J. Rexford, "Pretty good BGP: Improving BGP by cautiously adopting routes," in *ICNP*, 2006.

[9] S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure border gateway protocol (Secure-BGP) - realworld performance and deployment issues," in *NDSS*, 2000.

[10] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol(S-BGP)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, 2000.

[11] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Topology-based detection of anomalous BGP messages," in *RAID*, 2003.

[12] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A prefix hijack alert system," in *15th USENIX Security Symposium*, 2006.

[13] Y. Li and M. G. Gouda, "Balanced peer lists: Towards a collusion-resistant BGP," The University of Texas at Austin, UTCS Technical Report TR-09-16, 2009.

[14] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," in *Sigcomm*, 2002.

[15] R. Oliveria, B. Zhang, and L. Zhang, "Observing the evolution of internet AS topology," in *SIGCOMM*, 2007.

[16] A. C. Popescu, B. J.Premore, and T. Underwood, "Anatomy of a leak: AS9121," http://www.nanog.org/mtg-0505/pdf/underwood.pdf, May 2005.

[17] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," in *Sigcomm*, 2006.

[18] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," in *RFC 1771*, March 1995.

[19] G. Siganos and M. Faloutsos, "Neighborhood watch for internet routing: Can we improve the robustness of internet routing today?" in *Infocomm*, 2007.

[20] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos, "Power-laws and the as-level internet topology," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 514–524, 2003.

[21] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz, "Listen and whisper: Security mechanisms for BGP," in *NSDI*, 2004.

[22] T. Wan, E. Kranakis, and P. V. Oorschot, "Pretty secure bgp (psBGP)," in *NDSS*, 2005.

[23] R. White, "Securing bgp through secure origin bgp," *The Internet Protocol Journal*, vol. 6, no. 3, pp. 15–22, 2003.

[24] E. L. Wong, P. Balasubramanian, L. Alvisi, M. G. Gouda, and V. Shmatikov, "Truth in advertising: lightweight verification of route integrity," in *PODC*, 2007.

[25] B. Zhang, R. Liu, D. Massey, and L. Zhang, "Collecting the internet AS-level topology," in *ACM Sigcomm CCR Special Issue*, Jan. 2005.

[26] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "iSPY: Detecting ip prefix hijacking on my own," in *Sigcomm*, 2008.

[27] M. Zhao, S. W. Smith, and D. M. Nicol, "Aggregated path authentication for efficient bgp security," in *CCS*, 2005.

[28] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "An analysis of BGP multiple origin as (MOAS) conflicts," in *IMW*, 2001.

[29] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Detection of invalid routing announcement in the internet," in *DSN*, 2002.

[30] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A light-weight distributed scheme for detecting ip prefix hijacks in real-time," in *Sigcomm*, 2007.