# Safeguarding Data Delivery by Decoupling Path Propagation and Adoption

Mingui Zhang*
mingui.zhang@gmail.com

Bin Liu*
liub@tsinghua.edu.cn

Beichuan Zhang†
bzhang@arizona.edu

***Abstract*—False routing announcements are a serious security problem, which can lead to widespread service disruptions in the Internet. A number of detection systems have been proposed and implemented recently, however, it takes time to detect attacks, notify operators, and stop false announcements. Thus detection systems should be complemented by a mitigation scheme that can protect data delivery before the attack is resolved. We propose such a mitigation scheme, QBGP, which decouples the propagation of a path and the adoption of a path for data forwarding. QBGP does not use suspicious paths to forward data traffic, but still propagates them in the routing system to facilitate attack detection. It can protect data delivery from routing announcements of false sub-prefixes, false origins, false nodes and false links. QBGP incurs overhead only when there are suspicious paths, which happen infrequently in real BGP traces. Results from large scale simulations and BGP trace analysis show that QBGP is light-weight yet effective, and it converges faster and incurs less overhead than Pretty Good BGP.**

## I. Introduction

In the current global routing system, the contents of routing updates are not authenticated. False routing information may be announced into the Internet and accepted by other networks, causing problems such as service outages and eavesdropping. A special case is prefix hijacking, in which a network announces an IP prefix that belongs to another network. In a recent incident on February 24, 2008, AS 17557 announced one of YouTube's prefixes, diverting YouTube's traffic to AS 17557 and causing YouTube service outage worldwide for more than two hours [1]. Besides prefix hijacking, false paths towards the correct prefix origin can also be announced. Malicious attackers can use false routing announcements to hide their network identity in sending spams, inflict denial-of-service attacks by dropping victim's traffic, or even manipulate victim's traffic before forwarding it to the destination [2].

To deal with the problem of false routing announcements, several detection systems have been developed in recent years, including Cyclops [3], PHAS [4], MyASN [5], IAR [6], iSPY [7], Neighborhood Watch [8], origin list [9], and Lightweight Probing [10]. These systems detect false routing by examining routing updates, probing data paths, cross-checking with registry databases, or a combination of these techniques. Once a false routing case is detected, the owner of the prefix will be notified, and it is expected that the owner will take actions to resolve the problem, which, in today's Internet, usually involves contacting the offending network or its upstream provider to stop the false announcements. This process of detection, notification and resolution takes time, ranging from an hour to a day in past incidents and varying from network to network [8][1]. In the meantime, the damage to data traffic has already been made and malicious attackers may have already achieved their goals.

Therefore, a mitigation mechanism is needed for routers to protect the data traffic, *e.g.*, by not forwarding data along suspicious paths, before the attacks are resolved. On the one hand, the detection system needs a mitigation mechanism because data traffic is vulnerable for hours before the attack can be stopped. On the other hand, the mitigation mechanism also needs the detection system because identifying false routing is such a challenging task that a router cannot do it accurately with its limited information, resource and time. Thus effective routing defense needs detection and mitigation to complement each other.

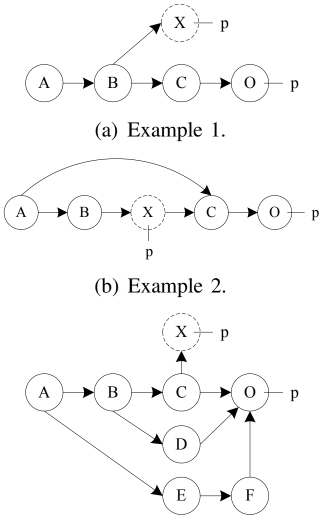
However, there is a dilemma: mitigation tries to render the attack ineffective while detection needs the attack to be effective in order to detect it! For instance, on September 22, 2008, a Russian ISP AS8997 hijacked a large number of prefixes as it leaked its routing table [11]. These false routing announcements were filtered (*i.e.*, dropped) by its upstream provider as a common mitigation practice. As a result, detection systems such as MyASN and IAR did not pick up this incident because the false announcements did not propagate to their monitors, and the owners of offended prefixes would not take any action since they were not aware of the incident. Meanwhile, ISPs and users within Russia were affected by the incident but could not detect or resolve it. Other more sophisticated mitigation schemes, such as PGBGP [12] and PurgePromote [13], also share the same problem of getting in the way of the detection system, both at control plane and data plane.

We propose to *decouple path propagation and path adoption* to ensure that mitigation mechanism and detection system can work together. In current BGP, the path a router adopts for data forwarding is the same path being propagated to neighbors. That is why upon receiving a suspicious path, a router has to either accept it (no mitigation but good detection)

*Tsinghua National Laboratory for Information Science and Technology, Computer Science and Technology Department, Tsinghua University, Beijing 100084, China.

†Computer Science Department, The University of Arizona, Tucson, AZ 85721, USA.

This work is supported by NSFC (60625201, 60873250, 60903182), 863 high-tech project (2007AA01Z216), 973 project (2007CB310701), Tsinghua Research Plan (Z02-2), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20060003058), and US DHS grant N66001-08-C-2028. This work was done during Mingui Zhang's visit to the University of Arizona when he was supported by the China Scholarship Council (2008621056) and partially by NSF award CNS-0721863.



(a) Example 1.

(b) Example 2.

(c) Example 3.

Fig. 1. Attack Examples (p is the prefix and X is the attacker.)

or reject it (good mitigation but no detection). Our idea is for a router to use trusted paths for data forwarding, but still inform its neighbors about the suspicious path. This way the data traffic is protected while the false routing announcements are being propagated to the detection system. Our design, dubbed "QBGP" (Q for quarantine), uses a simple rule to catch a wide range of suspicious announcements. The suspicious paths will be carried in an optional transitive attribute in BGP updates, while the routers still use trusted paths for data forwarding. Evaluation using large scale simulations and BGP trace analysis shows that QBGP can effectively protect data delivery from different types of false routing announcements. Compared with PGBGP, QBGP can correct its false positives faster and incurs less overhead.

## II. QBGP Design

False routing announcements can include false prefix origin, false link, false intermediate nodes, and false sub/super prefix. Such announcements can be caused by either inadvertent misconfigurations or malicious attacks. For the ease of exposition, we use "false path" to refer all false announcements, and "attacker" to refer the network that makes the false announcements regardless of their intention. We say a network falls victim to a false announcement if its traffic goes to the attacker after the attack but did not go to the attacker before the attack.

In typical mitigation schemes like PGBGP [12] and PGBGP++ [14], a router identifies suspicious paths and then block its propagation for a period of time $T_s$, usually set to one day. During this period, the router uses an alternative path to forward data. If the suspicious path turns out to be a real attack, it is likely to be withdrawn by the origin network within $T_s$ when the attack is resolved. If after $T_s$ the path is still in the routing system, the router will assume the path is legitimate and starts propagating it. The problem with PGBGP is twofold: the block of suspicious paths prevents detection systems from seeing the attack, and the quarantine time $T_s$ happens at every hop, making a suspicious but legitimate path to take a long time ($hopcount * T_s$) to reach every network.

QBGP addresses both problems by decoupling path adoption from path propagation. Take Figure 1(a) as an example. Before X launches the attack, the preferred data path is "ABCO-p." When X makes a false announcement of X-p to B, B will regard this new path as suspicious because it would divert traffic to an AS (X) that previously was not on the data path. B will store the suspicious path in its Adj-RIBs-In, but keep using the existing path in its Loc-RIB for data forwarding. At the same time, B re-announces its path (BCO) to A, attaching the new, suspicious path (BX) as an optional transitive attribute in the update message. Router A learns this suspicious path, stores it, and propagates it further to its neighbors. This way, the suspicious path is propagated to the Internet but is not adopted for data forwarding. Once the attack is stopped, the false announcements will be withdrawn from the routing system, *i.e.*, deleted or replaced in the Adj-RIBs-In. However, if after $T_s$ the suspicious path is still in Adj-RIBs-In, then it is regarded as a legitimate path. B will install it in Loc-RIB for data forwarding and also announce it to its neighbors.

### *A. Identifying Suspicious Paths*

For a given prefix p, a path is trusted if it has been staying in the Adj-RIBs-In continuously for at least $T_s$. All the nodes, links, and origins that appear in trusted paths are trusted components, and the set of them is denoted by $trusted(\mathrm{p})$. This set of trusted components is derived from current contents of all Adj-RIBs-In without using a database to store historical information like PGBGP does. Nodes, links, and origins that do not belong to $trusted(\mathrm{p})$ are said to be suspicious components for this particular prefix p. A new path is suspicious if it contains any suspicious component for its prefix. However, not all suspicious paths need to be explicitly quarantined. QBGP quarantines paths that satisfy the following condition:

- *A new path is quarantined if and only if it is suspicious, more preferred than other alternative paths, and contains an AS that is not in the current data forwarding path.*

If the new path is not better than alternative paths, it will not be able to divert any traffic. One may suggest that the attacker can first announce a less preferred path so that QBGP routers will take it as a backup path without suspicion, and then make the primary path fail to trick the router to use the false backup path. But in this case, if the attacker has the control of the primary path, it can already get the traffic without doing this. If the attacker does not have control of the primary path, it will not know when the primary path may fail and which backup path the router will choose, thus the attack will not be effective.

If the new path does not introduce any new AS on the data path, it is not quarantined since it does not divert any traffic. In Figure 1(b), when X launches an attack by announcing X-p, this path is not quarantined by B since B already sends its traffic to X. B will accept this path and announces it to A. Assuming ABX-p is more preferred than ACO-p, A will quarantine ABX-p since this new path would divert A's traffic to a new place, AS X, and X is a suspicious origin to A.

To reduce the potential false positives in quarantining paths, we introduce an optimization rule. If the current best path is replaced by a suspicious path (*i.e.*, the same neighbor that sent the best path previously sends another path to replace it), then the current best path is cached for a short time period. This rule is used to accommodate some unstable prefixes, which may get announced and withdrawn or oscillate between two paths frequently. With this rule, quick re-announcement of a path will not be treated as suspicious.

Previous measurement work has shown that (1) most prefixes are stable, and only a small number of prefixes are very unstable; (2) the most popular prefixes are stable; and (3) the most preferred paths are being used by routers for most of the time [15][16][17]. Therefore, we believe QBGP's criterion for suspicious paths will not generate excessive false positives in reality. Our evaluation using a regular week of BGP data traces (presumably without attacks) shows that 68.5%$\sim$74.8% of prefixes are not suspicious, and the majority of the rest prefixes is only suspicious infrequently.

### B. Choosing Alternative Paths

When a new path is the most preferred but suspicious, QBGP routers will use an alternative path for data delivery. The question is which alternative path to be chosen. First, if the existing path that is being used for data forwarding is still the best, then the router can stick to that path without any changes. Second, if the existing path in use will have been replaced by the suspicious path, then the router needs to pick an alternative. For example, in Figure 1(c), suppose C does not deploy QBGP and blindly accepts the false announcement X-p. B's existing path BCO-p will be replaced by a suspicious path BCX-p, therefore B needs to temporarily switch to a backup path BDO-p from its Adj-RIBs-In. Third, if there is no alternative path or all alternative paths are labeled as suspicious, then the router err on data delivery by adopting a suspicious path to forward packets.

### C. Propagating Updates

QBGP uses a new BGP attribute, QASPATH (Quarantined ASPATH), to carry the suspicious path in updates. QASPATH is defined as an optional transitive attribute. If a router does not understand this attribute, it will just pass it on to the next router, making QBGP incrementally deployable. In certain cases QBGP update may need to carry multiple QASPATHs. For example, in Figure 1(c), assume C does not deploy QBGP and accepts the false announcement of X-p. When B receives BCX-p, B quarantines this new path and switches to its backup BDO-p. The update from B to A will have BDO as the ASPATH, and BCX as the QASPATH attribute. However, since BDO is suspicious to A as well, A will quarantine BDO and switch to AEFO. Thus the update from A will contain two QASPATHs: ABCX and ABDO. Whether an update contains multiple QASPATHs depends on the topology and routing policy. In the worst case, the number of QASPATH in an update is the same as the AS hop count of the ASPATH. Given AS paths are usually 4 to 5 hops and rarely goes to more than 10 hops, we do not expect this will make QBGP message too large, and it is confirmed in our simulations.

### D. Releasing Quarantined Paths

If the quarantined paths are false announcements, it is likely that within $T_s$, the attack will be stopped and these paths being withdrawn from the routing system. In this case, there is no explicit release of the quarantined path. Just the upstream router will send an update with empty QASPATH attribute. If $T_s$ has passed and the quarantined path is still in the Adj-RIBs-In, then it is more likely that this is a legitimate path. The router will treat the path as a regular path and make it go through the path selection process. If the path turns out to be the most preferred one, it will be used for data forwarding and trigger routing updates to neighbor routers.

## III. Evaluation

We have implemented QBGP in SSFNet [18] to evaluate its effectiveness and overhead. The simulation is based on an Internet AS topology [19] with 23718 nodes and 94468 links, inferred AS relationship, and the "no-valley" and "customer-first" routing policy. We have also evaluated QBGP using BGP traces from RouteViews [20]. The ASes in the topology are classified into four types [21]: large ISPs, tier-1 ISPs, small ISPs, and stub networks. We randomly choose 50 ASes containing all the four types as attackers to inject false routing announcements. In each simulation run, one attacker AS X is active and it performs the following attacks[1]

- False Origin: X announces "X-p," where p is a prefix belonging to a different AS.
- False Node: X announces "XYO-p," where O is the real owner of the prefix p, but Y is an AS number made up by X.
- False Link: X announces "XO-p," where the link X-O is made up by X.
- False Sub-prefix: X announces "X-$p^+$," where $p^+$ is a sub-prefix of p.

We compare QBGP's effectiveness and overhead with PGBGP and PGBGP++. Table I summarizes the results. Overall QBGP is as effective as PGBGP++, incurs less communication and memory overhead, and converges faster. The rest of this section presents the details of the evaluation results.

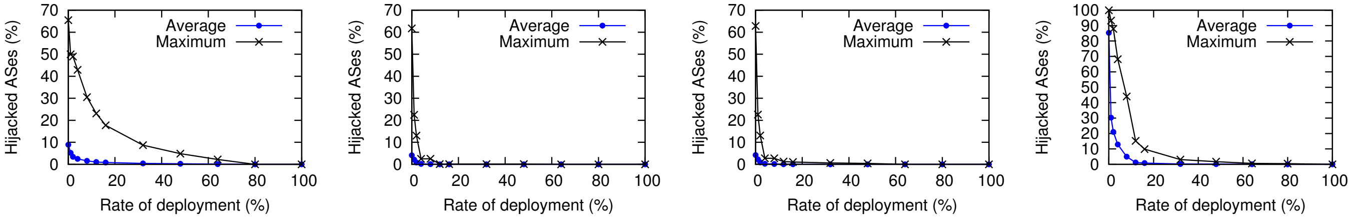
### A. Effectiveness Against Attacks

When the network converges after the false routing announcement, we examine the Loc-RIB of each AS router to determine whether it has fallen victim. With plain BGP, the average percentage of victim ASes are 8.91%, 4.10%, 4.20% and 85.31% for the four attack scenarios respectively. The actual number in each simulation varies depending on which AS is chosen as the attacker. When QBGP is fully deployed, no AS will take the false routes. PGBGP++ has the same effectiveness, while PGBGP only deals with false origin and false sub-prefix.

[1] False super-prefix attack is not included as it is equivalent to announcing a false origin of unused IP space.

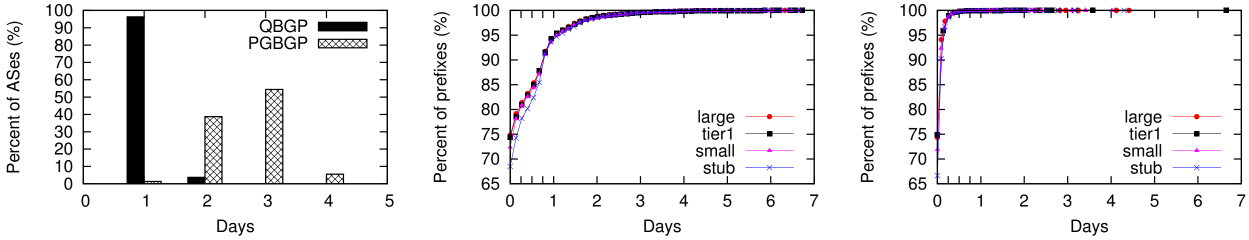
TABLE I
COMPARISON OF QBGP, PGBGP AND PGBGP++.

| | | PGBGP | PGBGP++ | QBGP |
|---|---|---|---|---|
| Coping with attack type | Origin changing | yes | yes | yes |
| | Node or link changing | no | yes | yes |
| Convergence time | Propagation time | $hopcount * T_s$ | $hopcount * T_s$ | $T_s$(96.28%) |
| | detour time | $hopcount * T_s$ | $hopcount * T_s$ | $<T_s$(95%) |
| Communication overhead | Extra updates/Original updates | $hopcount$ | $hopcount$ | $\sim$1 |
| | Extra octets | 0 | 0 | 0.15%$\sim$0.58% |
| Memory overhead | Extra memory | 20% | 40% | 3.25%$\sim$5.18% |
| | Historical database | yes | yes | no |



(a) False Origin (b) False Node (c) False Link (d) False Sub-prefix

Fig. 2. Effectiveness of partially deployed QBGP against different types of attacks



(a) The time before adopting new legitimate best path (b) The CDF of detour time per prefix in the first week of June 2008. $T_s$=24 hours. (c) The CDF of detour time per prefix in the first week of June 2008. $T_s$=4 hours.

Fig. 3. The convergence of QBGP

When QBGP is partially deployed, its effectiveness increases along with the deployment, the the most gain comes from the initial stage (Figure 2).

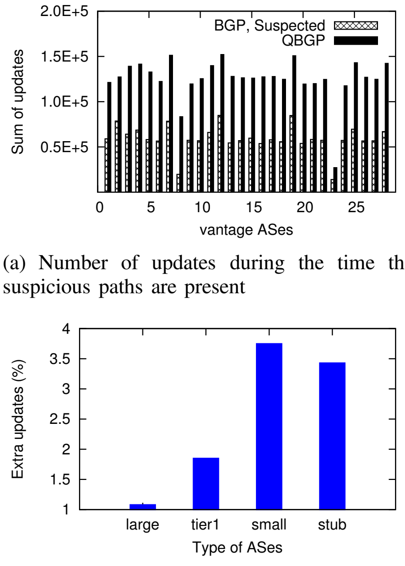
### *B. Convergence*

It is unavoidable that some legitimate paths will be deemed suspicious, *i.e.*, false positives, but a good mitigation scheme should be able to adopt the legitimate paths after only a short period of time. In QBGP, it is the quarantine time $T_s$. Since the paths have already been propagated, once $T_s$ has passed, routers should be able to adopt the paths right away. In PGBGP/PGBGP++, however, quarantine happens at every hop along the path. Therefore a router has to wait for $hopcount * T_s$ before adopting the legitimate paths. Figure 3(a) shows the simulation results using $T_s$ = 1 day. In QBGP, 96.28% ASes adopt the legitimate paths within one day, the rest within two days since they do not receive the path announcements in the first day due to BGP's poison reverse. In PGBGP/PGBGP++, most ASes take two to four days before adopting the legitimate paths.

We define *detour time* as the cumulative time that a router spends on using non-best path due to the path quarantine. Using the first week of June 2008 BGP trace data from RouteViews, we calculate the distribution of detour time over all prefixes. Figure 3(b) shows that more that 68.5%$\sim$74.8% prefixes have zero detour time, meaning that they do not see any suspicious paths over this week. The percentage of prefixes that experience detour time no more than one day is 95%, and this number increases to 99.7% if $T_s$ is set to four hours (Figure 3(c)). We have confirmed that the long tails in both figures are due to a small number of highly unstable prefixes as discovered in [17].

### *C. Message and Memory Overhead*

When a BGP router receives a new best path, it will send updates about this new path to its neighbors. However, if such a new best path is deemed suspicious by QBGP, a router will send out updates in two rounds: first is the alternative path with the QASPATH attribute, and the second is the new best path after the quarantine time $T_s$. Therefore for each false positive, the number of updates in QBGP should be roughly twice as that in BGP. This is confirmed by simulation results in Figure 4(a). However, since suspicious paths happen infrequently considering all BGP updates, the total number of updates in QBGP is only a few percent more than that in plain BGP. Figure 4(b) shows the result using one-month BGP traces in June 2008. The extra updates introduced by QBGP ranges from 1.08% to 3.75% for different types of ASes. Note that since PGBGP/PGBGP++ quarantine suspicious paths at every hop, their extra updates will be significantly more than QBGP's,



(a) Number of updates during the time that suspicious paths are present

(b) Extra updates of QBGP in a month

Fig. 4. Message overhead of QBGP

and should be proportional to the hop count of the topology diameter.

We also estimate the extra memory required to store the QASPATH attribute in the routing table. This extra memory is required only when suspicious paths are present, and over the month of June 2008, it's between 3.25% to 5.18% for different types of ASes. As a comparison, PGBGP/PGBGP++ maintains history information for all prefixes in a database all the time, which will cost much more router memory.

To summarize, QBGP incurs overhead only when suspicious paths are present, which happens infrequently in BGP traces. QBGP is effective in protecting data traffic in all the attack scenarios, incurs only small overhead in routing updates and router memory, and converges much faster than PGBGP/PGBGP++.

## IV. Related Work

Solutions to false routing announcements can be classified into three categories: prevention (*e.g.*, [22], [23], [24]), detection (*e.g.*, [5], [6], [7], [8], [9], [10], [4], [3], [25], [26]) and mitigation (*e.g.*, [12], [14], [13]). Among the mitigation schemes, PGBGP [12] and PGBGP++ [14] use a history database to identify suspicious paths and block their propagation. PurgePromote [13] purges the bogus routes and promotes valid routes at the same time to mitigate the impacts of attacks. These schemes have the side-effect of getting into the way of detection systems. QBGP addresses this problem by propagating suspicious paths but does not adopt them for data forwarding.

## V. Conclusions

Routing security of the future Internet will not be provided by a single mechanism; it is more likely to be a multi-line defense consisting of different mechanisms working before, during, and after attacks. QBGP provides effective protections for data delivery in face of ongoing false routing announcements. Compared with previous mitigation schemes, QBGP reduces the delay of legitimate announcements significantly, and only incurs a small amount of communication and memory overhead. More importantly, QBGP is complementary to existing prevention and detection systems, making it possible for them to work together for better routing security.

## References

[1] [Online]. Available: http://www.ripe.net/news/study-youtube-hijacking.html

[2] H. Ballani, P. Francis, and X. Zhang, "A Study of Prefix Hijacking and Interception in the Internet," in *Proceedings of ACM SIGCOMM*, 2007.

[3] Y.-J. Chi, R. Oliveira, and L. Zhang, "Cyclops: the as-level connectivity observatory," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 5, pp. 5–16, 2008.

[4] M. Lad, D. Massey, D. Pei, B. Zhang, and L. Zhang, "PHAS: A Prefix Hijack Alert System," in *15th USENIX Security Symposium*, 2006, pp. 153–166.

[5] "RIPE myASN System," http://www.ris.ripe.net/myasn.html.

[6] [Online]. Available: http://iar.cs.unm.edu/

[7] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "iSPY: Detecting IP Prefix Hijacking on My Own," in *Proceedings of ACM SIGCOMM*, 2008.

[8] G. Siganos and M. Faloutsos, "Neighborhood Watch for Internet Routing: Can We Improve the Robustness of Internet Routing Today?" in *Proceedings of IEEE INFOCOM*, 2007.

[9] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Dection of Invalid Routing Announcement in the Internet," in *Proceedings of the IEEE DSN*, June 2002.

[10] 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 *Proceedings of ACM SIGCOMM*, 2007.

[11] "Prefix hijack by ASN 8997." [Online]. Available: http://www.merit.edu/mail.archives/nanog/2008-09/msg00704.html

[12] J. Karlin, S. Forrest, and J. Rexford, "Pretty Good BGP: Improving BGP by Cautiously Adopting Routes," in *Proceedings of IEEE ICNP*, 2006.

[13] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao, "Practical Defenses against BGP Prefix Hijacking," in *Proceedings of ACM CoNEXT*, 2007, pp. 1–12.

[14] J. Karlin, S. Forrest, and J. Rexford, "Autonomous security for autonomous systems," *Computer Networks*, vol. 52, no. 15, pp. 2908–2923, 2008.

[15] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "BGP Routing Stability of Popular Destinations," in *Proceedings of ACM IMC*, 2002, pp. 197–202.

[16] K. Butler, P. McDaniel, and W. Aiello, "Optimizing BGP Security by Exploiting Path Stability," in *Proceedings of ACM CCS*, Alexandria, VA, United States, 2006, pp. 298–310.

[17] R. V. Oliveira, R. Izhak-Ratzin, B. Zhang, and L. Zhang, "Measurement of Highly Active Prefixes in BGP," in *Proceedings of IEEE Globecom*, 2005.

[18] "SSFNet." [Online]. Available: http://www.ssfnet.org/homePage.html

[19] Y. He, M. Faloutsos, S. V. Krishnamurthy, and M. Chrobak, "Policy-Aware Topologies for Efficient Inter-Domain Routing Evaluations." in *Proceedings of IEEE INFOCOM*, 2008.

[20] "University of Oregon Route Views Project." [Online]. Available: http://www.routeviews.org/

[21] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "Quantifying the Completeness of the Observed Internet AS-level Structure," UCLA CS Department, Tech. Rep. 080026, Sept 2008.

[22] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582 – 592, 2000.

[23] J. Ng, "Extensions to BGP to Support Secure Origin BGP," April 2004, ftp://ftp-eng.cisco.com/sobgp/drafts/draft-ng-sobgp-bgp-extensions-02.txt.

[24] Y.-C. Hu, A. Perrig, and M. Sirbu, "SPV: Secure Path Vector Routing for Securing BGP," in *Proceedings of ACM SIGCOMM*, 2004.

[25] J. Qiu, L. Gao, S. Ranjan, and A. Nucci, "Detecting bogus BGP route information: Going beyond prefix hijacking," in *Proceedings of SecureComm*, 2007, pp. 381–390.

[26] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Topology-based Detection of Anomalous BGP Messages," in *Proceedings of RAID*, 2003, pp. 17–35.