Securing BGP Through Existing Infrastructure and Contractual Chains (CCBGP)

Yuri Cantor, Nancy Griffeth, and Bilal Khan
Department of Computer Science
The CUNY Graduate Center
365 Fifth Avenue New York City, NY 10016, USA

June 17, 2010

Abstract

This paper proposes a novel approach that draws upon the existing architecture and contractual relationships and compares the approach to the main existing techniques for securing BGP against prefix hijacking. We define prefix hijacking as usurping control of IP prefixes through the manipulation of BGP routing tables resulting in a redirection of network traffic away from a *correct* route to a prefix, which traverses all and only the ASes in the route advertisements and abides by BGP policy finally terminating at the AS that owns that prefix route, and onto another route. We further refine our definition to not include those attacks where the attacking AS lies along the correct path but does not actually route packets as it advertises. Our novel approach, termed Contractual Chained BGP, completely eliminates prefix hijacking under certain plausible assumptions[1] and provides support for accountability in the form of forensic traceback. CCBGP applies contracts to build a transient chain of links between AS neighbors and neighbors of neighbors. Because the links are transient, each AS need only be aware of the links in its contractual sphere. Keeping the contractual sphere small limits the computational requirements of creating a chain link while the overlap of the links provides the security of a complete chain.

1 Introduction

The security of the protocols that enable the Internet to function have become essential to the financial, political, and commercial functioning of society. The development of these protocols did not anticipate the current threats, and exactly because security was once a non-issue it is now paramount. This paper seeks to address one well-known and serious BGP vulnerability, prefix hijacking[5].

Specifically this paper borrows from ideas applied in other areas of security to substantially increase the difficulty of prefix hijacking, to provide forensic trace back to the originator of the hijack, and to connect these directly to the relationships between neighboring autonomous systems. Previous recommendations for securing the same BGP vulnerability have relied on costly modifications to the infrastructure and protocol, and as a result these recommendations have not been deployed widely. In contrast this paper proposes a method for defending against prefix hijackings with only a moderate cost for deployment and a seamless integration with the existing infrastructure and protocol. The proposed innovation is to utilize the existing contractual relationship to form decentralized links which when combined create of a chain of trust. The chain of trust can be used both to prevent prefix hijackings and to trace back any attempts to hijack.

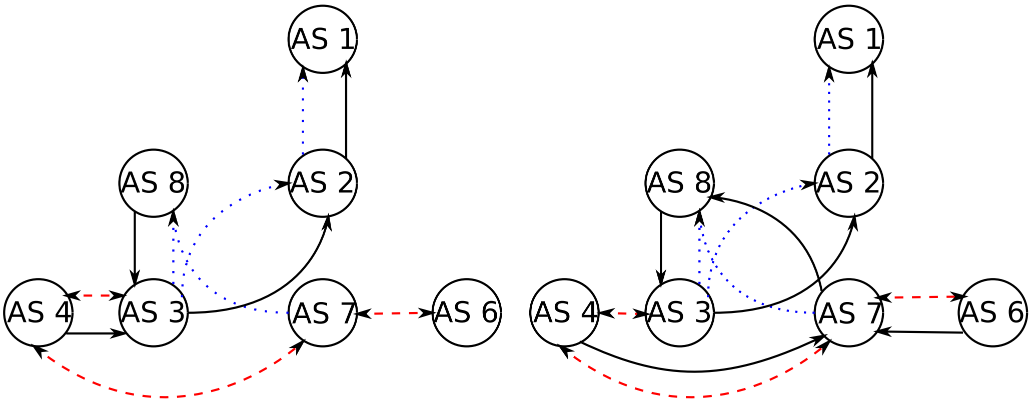
[1] Under the assumption of non-collusion between direct neighbors and protocol assumptions which are elaborated in section 4.2.

Additionally, the nuance of utilizing the contractual relationship in building the links of the chain allows for legal recourse.

2 How BPG works and what makes it vulnerable

The Internet is made up thousands of networks called Autonomous Systems (ASes). A single institution controls each AS, though some institutions control more then one AS. ASes exchange routing information with each other using a routing protocol called Border Gateway Protocol (BGP). Within an AS, an Interior Gateway Protocol (IGP) handles routing and routing information is shared between border routers using internal Border Gateway Protocol (iBGP). Routers combine information from both IGP and BGP to create a forwarding table. This dependence upon exchanged routing information to create a dynamic forwarding table is precisely what makes BGP, and routing protocols in general, susceptible to attacks. In BGP, routing information is exchanged using update messages[2]. Each update message can contain route advertisements, route withdrawals, or keep-alive messages. The attacks this paper focuses on involve the route advertisements. BGP uses policies to determine which route advertisements to send and which to accept. When a BGP router has several possible routes to a particular destination, path attributes break the tie [7]. Using the knowledge of how ties are broken, an attacker can craft an advertisement containing the attacker's preferred path to an existing destination and cause a BGP router to update its forwarding table with a path chosen by the attacker.

The relationships between these ASes can be described by three roles: customer, peer, and provider. These relationships govern the routing policies between the ASes and consequently the contractually agreed upon flow of traffic. However, an institution can (and often will) manipulate the flow of traffic using the known BGP update preferences. The mechanisms in place are not designed to prevent this, but rather to prevent basic routing problems like loops (by rejecting routes advertisements which already contain the AS's AS number as part of the path), route flapping (using dampening), and basic masquerading (using authentication). BGP's policies can filter exported and imported routes as well as re-write path attributes.



(a) Normal

(b) Attack

Figure 1: Route Flow Trees (Red Dash=Peer to Peer, Blue Dot=Provider to Customer, Black Line=Route to Prefix Originator)

In Figure 1a, AS 1advertises that it originates a prefix. AS 2 as the provider for AS adopts its customer's

[2] Route withdrawal messages also contain information and could result in attacks. These attacks could be handled using time outs of route validity along with signatures, but are not addressed specifically in this paper.

route and propagates it to AS 3. AS 3 as the provider for AS 2 adopts its customer's route and propagates it to both AS 4 and AS 8. AS 4 as a peer provides connectivity for AS 3's customers and propagates it to its customers. Note: AS 6 and AS 7 would not receive the advertisement.

In Figure 1b, AS 1advertises that it originates a prefix AS 2 as the provider for AS 1 adopts its customer's route and propagates it to AS 3. AS 3 as the provider for AS 2 adopts its customer's route and propagates it to both AS 4 and AS 8. AS 8 maliciously advertises to AS 7 that it originates the Prefix. AS 7 as the provider for AS 8 adopts its customer's route and propagates it to AS 4 and AS 6. AS 4 as a peer provides connectivity for AS 7's customers and propagates it to its customers. AS 6 as a peer provides connectivity for AS 7's customers. AS 4 prefers the route from AS 7 over AS 3's because it is shorter. Once this happens, man-in-the-middle attacks from AS 8 are possible.

3 Proposed BGP Security Additions

There have been many proposed additions and modifications to BGP that are intended to provide security against the above-described known vulnerability. This section surveys in chronological order the main proposals for securing BGP and details their incremental improvements and novel additions. The proposals described in this section are as follows: Secure BGP (SBGP) proposed in 1996, Secure Origin BGP (SOBGP) proposed in 2002, Secure Path Vector Routing for securing BGP (SPV) proposed in 2004, Pretty Secure BGP (PSBGP) proposed in 2005, and Pretty Good BGP (PGBGP) proposed in 2006.

3.1 SBGP

SBGP proposes implementing a Public Key Infrastructure (PKI) and using additional update messages, termed attestations, which would contain routing updates signed with an AS's private key. SBGP relies on IPSec to provide integrity and authentication, and uses certificates from a PKI to verify particular update messages were originated by a particular AS. SBGP adds signatures to all messages. The certificates used in combination with the signature provide ASes with a method of checking the authenticity and integrity of the message. Delegation messages are used to demonstrate one AS allowing another AS to advertise a particular route. Attestations contain route attestations and address attestations and could be carried in BGP via a new optional path attribute. The attestations work by forming a chain of signatures for each AS along the path.

SBGP uses a PKI and consequently, requires additional infrastructure. The attestation acts as an additional update packet. And, additional storage is needed to maintain the public key and private key pair. Finally, additional cryptographic computation is required to verify the signatures and to sign the attestations.

SBGP does a good job limiting prefix hijacks but faces problems with tunneling attacks to draw in traffic via collusion. It also faces severe performance limitations due to the overwhelming computational cost of creating a chain of attestations from a prefix originator to the end of a path [4].

3.2 SOBGP

SOBGP's novel contribution addresses the SBGP's use of a centralized PKI. The move is from centralized to decentralized while maintaining the basic functionality of SBGP and its signature chains.

SOBGP builds off of SBGP and focuses on the following four points: authorization to advertise origination of a prefix, existence of a path from the advertising AS, authorization by the originating AS for another AS to advertise a route to the prefix, and path correctness with regard to policies. SOBGP uses entity certification via a "web of trust" in the place of a centralized key authority and trusted third parties (a PKI) as a foundation binding most of the ASes to their corresponding public key. SOBGP uses the hierarchical ownership of blocks of IPs in conjunction with certificates using signatures from trusted third parties to extend trust outwards on the ownership tree. For example, the signature by a trusted third party is used to validate the signature of an owner of a block of IPs. That owner is then able to use sign certificates for any subset of his owned block of IPs. In other words, if an AS has a third party signature for its certificate then that AS will become trusted to generate signatures for certificates.

These certificates can then be used to grant permission to an AS to advertise a route via a signature from the IP block originator. As permission is granted to advertise, ASes can derive the network topology using these advertisements which are signed. Each AS has signatures from all peers that they are, in fact, neighbors. Each AS can use topological advertisements and in turn verify both the path and that the path was permitted.

SOBGP adds signatures for each attestation and delegation, i.e. messages are signed by a prefix originator permitting another AS to advertise a route to that prefix. The signatures require additional infrastructure in the form of a PKI and additional update packets that contain the signatures as well as the additional approvals. The keys for these signatures require additional storage and the signatures themselves require additional cryptographic computation. Each update message requires computation in the verification of the signature and checking that the appropriate prefix originator has approved the route update.

While not absolutely preventing every possible hijack, SOBGP offers strong protection from prefix hijackings by applying the learned topological map. SOBGP provides a method for verifying if an advertisement was permitted, but does not provide a method for verifying that a prefix was hijacked. Consequently, SOBGP does not provide a method for tracing the prefix hijacker nor does it offer recourse for the hijacking itself [8].

3.3 SPV

SPV's novel contribution addresses the cost of SBGP's public key cryptographic operations by introducing symmetric keys to supplement an initial exchange. However, SPV continues to use the PKI overlay structure just as SOBGP and SBGP do as well as the signature chains.

SPV can use short-lived keys that are generated offline for one time signatures, thereby increasing security without increasing the computational cost dramatically. ASes using SPV forward the attestations containing signed advertisements so that their subsequent route advertisements can be verified using the forwarded attestation.

SPV uses additional infrastructure including offline workstations to handle the key generation and mapping. SPV also uses additional update packets to act as the attestations do in SBGP; consequently, additional storage is needed for the keys and additional cryptographic computation is needed for the verification of the attestations.

SPV does not completely prevent hijacks, but it limits them in the same way that SBGP and SOBGP do. SPV does not identify that hijacks occurred nor does it identify the hijacker. Nor does SPV offer any recourse for the hijack [2].

3.4 PSBGP

PSBGP builds off of SBGP and SOBGP by applying a decentralized trust model for verification of IP prefix ownership. PSBGP continues to use the centralized trust model to authenticate AS numbers and AS public keys. Because of a centralized starting point for verifying the keys and AS numbers PSBGP offers stronger protection from impersonation then the decentralized model with a web of trust. On the other hand, PSBGP applies a decentralized model for prefix ownership verification which is weaker against prefix hijacking. PSBGP seeks to find a balance between the cost of SBGP's security operations and the security guarantees. Instead of using signed attestations as SBGP does, PSBGP uses a number of its peers to endorse its advertisements. PSBGP requires additional infrastructure in the form of a centralized PKI which maps an AS to a public key. PSBGP requires additional update packets to endorse the advertisements of peers. Additional storage is required for the public and private key pair along with additional cryptographic computations for verifying the signatures on the endorsement packets.

PSBGP's additional security does not completely prevent hijacks but it does limit them through the endorsement packets. PSBGP does not have any method for identifying that a hijack occurred nor does it identify the hijacker. And, PSGBP offers no recourse for hijacks [6].

3.5 PGBGP

PGBGP takes a novel approach that gives up signature chains and PKIs and instead uses a delay in applying the changes advertised in an update message. The delay relies on existing redundancy in the Internet to avoid denial of service. This delay provides an AS time to verify the route advertisement.

PGBGP limits prefix hijackings by learning origins of prefixes and building a route table over a training period. Then changes to the routing table and origins are quarantined for a period of time, a suspicious period, while the existing redundant routes are used. If the routes persist after the suspicious period elapses, then the new route is accepted. By cautiously accepting new reachability information and investigating and repairing bogus routes PGBGP mitigates through a delay the effects of hijacks making them more difficult to time and deploy. However, PGBPG relies on redundancy in existing paths; otherwise, delays in accepting new routes could cause a denial of service.

PGBGP does not require any additional infrastructure. Instead it requires some kind of supervision and a training period. Any redundancy required by PGBGP is assumed to be a part of the network infrastructure and consequently is not additional. No additional update packet is needed; rather the security derives from the delay in accepting and applying the information in the update packet. During the delay, the routes can be verified by administrators or via some other process. There is no additional storage or cryptographic computation necessary. However, the non-automated cost is neither comparable to the cost of the other methods nor acceptable in a fully autonomous dynamically updating routing protocol.

The hijacks are not prevented though they can be limited by PGBGP's delay in accepting route updates. PGBGP does nothing to identify that a hijack has occurred nor who the hijacker is. Further if offers no recourse for the hijack [3].

4 Contractual Chained BGP (CCBGP)

CCBGP builds on current proposals for securing BGP while limiting the overhead both in additional architecture and in computation. CCBGP builds a transient signature chain from pairs of signatures that form the links of the chain. Consequently, each AS is only responsible for verifying two signatures before it accepts the advertisement as opposed to protocols like SBGP needing to verify a signature for each AS in the route. CCBGP decentralizes the binding of key to AS number and uses the security provided by the direct contractual link from AS to AS when it binds the key. In contrast, SOBGP decentralizes binding keys to AS number using a trust relationship rather then a contractual relationship. CCBGP's chain links do not extend past the neighbor of a neighbor, avoiding the need for continual extension of trust down the routing path. In addition, CCBGP utilizes the contractual relationship to provide recourse for attacks. For detection of attacks CCBGP relies on existing techniques including ISPY and a Light Weight Distributed Scheme for Detecting IP Prefix Hijackings in Real Time (LWDS)[10]. These two complementary techniques if implemented correctly should detect most prefix hijacks. Lastly, CCBGP can operate seamlessly on top of BGP either by delaying acceptance of updates, in the same way PGBGP operates, or by reverting BGP updates that are accepted but not verifiable.

4.1 Motivation

The goal of CCBGP is to protect the AS itself from the attack, enable ASes to distinguish between verified and unverified advertisements, and to identify which ASes were involved in the hijack. Since CCBGP can run seamlessly on top of BGP running CCBGP will not negatively impact the current BGP interactions; it will only add more security with very little overhead.

4.2 Assumptions

For the implementation and evaluation of CCBGP we assume that the following:

- Direct neighbors of an AS can be and are verified out of band.[3]
- Anyone can verify an AS's neighbors.
- CCBGP keys of neighbors and keys of neighbor's neighbors can be verified out of band.[4]
- The contract between neighboring ASes is legally binding and carries severe penalties for involvement in prefix hijackings.[5]
- Logs of CCBGP updates are timestamped and paired with BGP updates and stored for a time frame specified in the contract and shared during trace back, or else the AS not sharing is can be held accountable.
- An AS can and does securely download from IANA a database that contains mappings of all ASes to their owned Prefixes.
- ISPY combined with LWDS will detect all types of prefix hijacks with very high probability.
- The cryptographic techniques applied to generate the signatures and public/private key pairs are secure.

4.3 CCBGP Steps

The proposed protocol is composed of five steps:

1. Every AS generates a public/private key pair on its own, and publishes its public key.
2. A legally binding contractual relationship with all neighbors is initiated or re-established in which the terms of service include: an agreement to sign only verified update packets under some heavy and a severe penalty if unverified packets are signed. Each AS exchanges a list of its neighbors and their keys, with each of its neighbors. This list is maintained throughout the relationship. These must be direct neighbors and must be verified out of band.
3. Every AS regularly updates and maintains a table of Prefix owners from IANA.[6]
4. Update messages that offer a new prefix originator are matched directly against the table maintained in step three. Any non-match is rejected. Update messages signed with the private key of the immediate neighbor and the immediate neighbor's neighbor are accepted and when propagated, should be propagated signed with the propagating AS's private key and its direct neighbor's signature but not

[3] Collusion with distant neighbors will be prevented because an AS would be unable to claim a neighbor it was not directly connected to.

[4] Prevents co-opting a neighbor's key because an AS would be unable to falsify a neighbor's key.

[5] Contracts will likely have to be updated to include the clause covering penalties for involvement in prefix hijackings.

[6] Collusion with distant neighbors will be prevented because an AS would be unable to claim a neighbor it was not directly connected to.

the neighbor's neighbor's signature. (in other words, just two signatures). Update messages not signed are either rejected or tentatively accepted with the understanding that they are not verified.

5. Pre-existing BGP routing and forwarding tables are checked for paths that are verified/verifiable and the unverifiable paths are verified through other means.

The actual implementation of these steps at each AS gateway router is up to the owner. Some ASes might wish to verify the neighbor's keys out of band or only accept them once or require some verification in order to advertise a new neighbor and the new neighbor's key. The AS gateway routers could accept all BGP updates and use CCBGP to review them, verifying those it can verify and reverting all unverified changes. Alternatively, BGP updates might face a delay in implementation during which time the unverified BGP advertisements are tested/checked for hijackings. The More lenient or tolerant implementations would enable CCBGP to operate seamlessly even if sparsely deployed. Where as a stricter implementation at tier 1 ASes might force quicker and broader deployment to prevent isolation of ASes. However, all CCBGP should provide a seamless integration with BGP while allowing each AS to determine how it wants to handle unverified advertisements (both in BGP and CCBGP in the case of neighbor's neighbor's keys).

4.4 CCBGP Pseudocode

CCBGP should run as an agent on each AS's BGP gateway router, examine the routing and forwarding tables, view incoming update packets, communicate with neighbors who establish a connection, and provide regular signed updates.

```
/ / intialization
  Create a copy of the route table (or enable the ability to revert to the original route table)
  Create table from IANA relating prefix originator to AS
  Create table relating public keys to AS for all neighbors and all of their neighbors

/ / Intermittent - every day (or as appropriate)
  Update table from IANA relating prefix originator to AS
  Update table relating public keys to AS for all neighbors and all of their neighbors
  Send an update of your public key and your neighbors public keys to your neighbors

/ / Processing Attestations and Update packets
  For each update packet
    Check originator against the table mapping AS to prefix originator
      Discard all faulty advertisements / / Break out of loop if discarded
    Listen for signed update packet for 1 epoch / / Before timing out
    If a signed update packet is received
      Allow BGP route tables update to remain in place
    Else
      If AS sending message uses CCBGP and another route to this prefix exists
        Revert the update
      If AS sending message does not use CCBGP and no other route to this prefixs exists
        Tentatively accept the update
      If AS sending message uses CCBGP but no other route to this prefix exists
        Tentatively accept the update
    Update backup route table/reversion table and forwarding table

/ / Sending update packets
  Send BGP update packet
```

Send CCBGP update packet

4.5 Proof

AS$_p$= The AS originating the Prefix
AS$_t$= Target AS, where the target is the recipient of the attacking route advertisement
AS$_a$= Attacking AS, the the attacker is the sender of the false route advertisement to a non-colluding AS.
AS$_{sa}$= Supporting Attacker AS, an AS that provides a supporting signed advertisement enabling ASa to convince ASt that the advertisement is legitimate.

We leave as a future area of research detection and prevention of a set of prefix hijacks which involve an attacking AS which lies along the correct path but does not actually route packets as it advertises. We view this type of attack as a trivial man in the middle attack rather then a prefix hijack because any node along a correct path to AS$_p$ would already have access to the traffic between a target AS and AS$_p$ without needing to change the route.

Assume all ASes have installed CCBGP or alternatively, assume we all paths traverse only ASes which are running CCBGP. We leave as a future area of research the benefits of CCBGP where paths traverse both ASes which are and are not running CCBGP.

Assume an attack has occurred involving AS$_p$ and that AS$_t$ detects this attack using ISPY or LIDS. Then AS$_t$ believes an attack has occurred involving a path to AS$_p$. AS$_t$ extracts all routes to AS$_p$ from the routing table and any recently logged routes to AS$_p$. Because an attack occurs AS$_t$ traces through all extracted routes. From each AS on each path, AS$_t$ requests the two corresponding signed packets for the route being traced. If an AS is unable to produce both packets, then that AS is an attacker. Because an attack has occurred on a path to AS$_p$ and because a successful attack will result in a new route entry, then AS$_a$ and AS$_{sa}$ will be queried for packets corresponding to the path they successfully hijacked. Because each AS on the extracted paths being queried will be forced to the produce two corresponding packets for the suspect route and because neither AS$_a$ nor AS$_{sa}$ can produce the two packets for the route being hijacked, then both AS$_a$ and AS$_{sa}$ will be identified as attackers. Therefore, if an attack is detected by AS$_t$, then AS$_a$ and AS$_{sa}$ will be identified as attackers. In other words, if an attack is detected, then all the attackers will be identified.

While the forensic trace back should occur on a separate system then the one handling the routing, the work should still be minimized. However, the challenge of using timing or other information as parameters to minimize the overhead of searching logs, querying neighbors for corresponding packets, and the number of comparisons of signatures necessary is left as future work.

4.6 Anomaly Correction

After detecting a prefix hijack and identifying the attackers, an AS should remove the hijacked route from its routing and forwarding tables and inspect other routes advertised by the hijacking AS. The AS may either revert to redundant routes or possibly experience an appropriate loss of connectivity to the prefix.

4.7 Forensics

The proposed idea only enables the ability to identify who originated a particular prefix advertisement, not that the prefix was hijacked. Consequently the proposed addition should be used in conjunction with other forensic techniques that identify that a hijack has occurred. ISPY can be deployed to detect hijackings which result in cuts, severed reachability for a particular prefix, and LWDS should be deployed to detect modifications to the path which do not result in cuts. Further, IANA should be used to regularly generate a table that maps ASes to their owned IP Prefixes.

4.8 CCBGP Summary

While CCBGP requires no additional infrastructure, CCBGP does require additional update packets containing signatures of the updates. And since a public and private key pair are used along with a secondary routing table, additional storage is also required. Signing update messages and verifing signatures require additional cryptographic computation. CCBGP does not prevent hijacks but it limits them to the case where two adjacent ASes are collaborating. And, if a hijack occurs CCBGP will detect it using ISPY or a LWDS. Next, CCBGP can use the signature chain to trace back and identify the hijacker. Finally CCBGP provides legal recourse for the hijack via the contractual agreements between the ASes.

5 Comparison to Other techniques

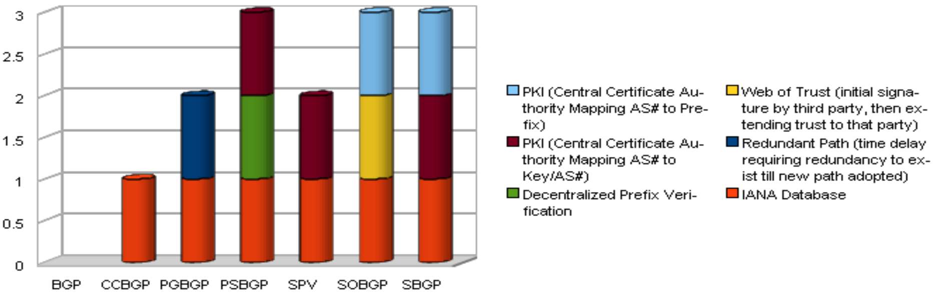


Figure 2: Infrastructure Requirements (Y-axis: Number of Infrastructures, X-axis: Protocol, Color: Type of Infrastructure)

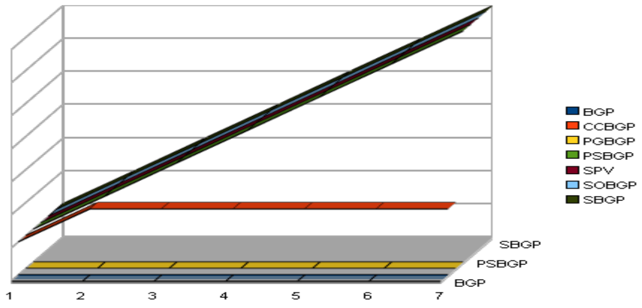


Figure 3: Signature Requirements (Y-axis: Number of Signatures Verified by an AS, X-axis: Route Length, Z-axis: Protocol)

In the figures we look at two features of BGP security protocols: infrastructure and cryptographic signatures. Figure 2 Compares BGP security protocols based upon additional required infrastructure while Figure 3 compares BGP security protocols based upon the number of signatures that must be verified by an AS for routes of varying length [9][1].

5.1 Similarities

CCBGP is similar to the SBGP, SOBGP, PSBGP, and SPV in that it uses cryptographic techniques to build chains of signatures. It is also similar to PGBGP in that it can use a delay to accepting updates that are not verified. CCBGP also applies a concept similar to the web of trust in SOBGP and PSBGP. This paper does not focus on techniques to optimize cryptographic efficiency; however, CCBGP could also use SPV's enhancement of the symmetric keys to reduce cryptographic costs.

Some problems may emerge in CCBGP that would be similar to those affecting other proposed techniques. Specifically, a DOS attack could arise from delays and rejections of updates/routes. Future work would explore simulating or deploying CCBGP and testing if novel attacks emerge.

5.2 Differences

The main advantage of CCBGP over the other techniques is that there is no reliance on additional infrastructure PKIs and there is no need for massive chains or extended lookups because CCBGP leverages the existing contractual relationships and the existing infrastructures while limiting the length of the signature chains to two.

CCBGP addresses the problem of trust (having a centralized authority like IANA or Identity Based Cryptography's central authority controlling communication) in an inherently decentralized network like the Internet. CCBGP does rely in part on the centralized structure of IANA but only to the extent that IANA is already central to the function of the Internet.

5.3 Protection against Side Channel Attacks

Often overlooked in protocol security are methods of attacks that are not directly associated with the protocol. CCBGP offers limited protection against one form of these side channel attacks: those directed at personnel. Soft vulnerabilities (vulnerabilities involving personnel e.g. bribery, blackmail, coercion, deceit, etc) are generally easier targets than hard vulnerabilities (vulnerabilities involving the physical and technical e.g. breaking encryption, buffer overflow, tapping a physical cable, etc). Strengthening the personnel target by requiring more then one person to hijack a prefix will both raise the cost of compromising the target personnel and increase the likelihood of being detected and caught. The properties of CCBGP enable trace back and provide legal recourse for the hijack thereby deterring potential personnel from turning into attackers. This dissuasive action raises the cost to compromise personnel and thus, raises the cost for this type of attack.

6 Conclusions

The paper proposes a novel approach to securing BGP against prefix hijackings, which has a comparatively low computational overhead and requires no new additional infrastructure. The lack of drawbacks and the increasing demands of security should motivate institutions administrating ASes to update their contracts to include a clause for penalties for involvement in a hijacking. This addition coupled with basic existing forensic software and a simple CCBGP agent on the AS gateway routers should suffice to securing BGP against most prefix hijacks as well as provide detection of those hijacks not protected against. Further, the penalties should act as a deterrent both to personnel and to institutions considering partaking in any hijack.

However, even with the implementation of CCBGP there are still many tools that can help improve security. Specifically, research should continue in how contracts and relationships can be leveraged for security. Answers to the following questions could provide guidance to where security resources are most effectively deployed: Who wants to hijack a prefix versus who actually hijacks prefixes, Why are prefixes hijacked, Which prefixes are hijacked most often, Which tier of AS usually initiates the hijacks, At which tier is security against hijacks most important, Where is collusion is more likely, and What tools can be used or created to enhance security and testing?

References

[1] C. Ellison and B. Schneier. Ten risks of pki: What youre not being told about public key infrastructure. In *Computer Security Journal, V 16, n 1*, pages 1–7, 2000.

[2] Y. Hu, A. Perrig, and M. Sirbu. Spv: secure path vector routing for securing bgp. In *SIGCOMM '04. ACM, In Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications (Portland, Oregon, USA, August 30 - September 03, 2004)*, pages 179–192, 2004.

[3] J. Karlin. Pretty good bgp: Improving bgp by cautiously adopting routes. In *In Proc. International Conference on Network Protocols*, 2006.

[4] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol (s-bgp). In *IEEE Journal on Selected Areas in Communications 18, 4 (Apr.)*, pages 582–592, 2000.

[5] M. Lad, R. Oliveira, B. Zhang, and L. Zhang. Understanding the impact of bgp prefix hijacks. In *ACM SIGCOMM. Poster*, 2006.

[6] P. v. Oorschot, T. Wan, and E. Kranakis. On interdomain routing security and pretty secure bgp (psbgp). *ACM Trans. Inf. Syst. Secur.*, 10(3):11, 2007.

[7] Y. Rekhter and Li. A border gateway protocol 4 (bgp 4). In *IETF RFC 1771*, 1995.

[8] R. White. Securing bgp through secure origin bgp. In *The Internet Protocol Journal 6, 3*, pages 12–22, 2003.

[9] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao. Practical defenses against bgp prefix hijacking. In *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, pages 1–12, New York, NY, USA, 2007. ACM.

[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 *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 277–288, New York, NY, USA, 2007. ACM.