Enhancing the Trust of Internet Routing with Lightweight Route Attestation

Qi Li†, Mingwei Xu†, Jianping Wu†, Xinwen Zhang‡, Patrick P. C. Lee§, Ke Xu†
†Dept. of Computer Science, Tsinghua University, Tsinghua National Lab for Information Science and Technology
‡Huawei America Research Center §Dept. of Computer Science and Engineering, The Chinese University of Hong Kong
†{liqi,xmw,jianping,xuke}@csnet1.cs.tsinghua.edu.cn ‡xinwen.zhang@huawei.com §pclee@cse.cuhk.edu.hk

ABSTRACT

The weak trust model in Border Gateway Protocol (BGP) introduces severe vulnerabilities for Internet routing including active malicious attacks and unintended misconfigurations. Although various secure BGP solutions have been proposed, they share similar weaknesses such as high complexity of security enforcement and incapability of data-plane attack prevention. We propose TBGP, a trusted BGP scheme aiming to achieve high authenticity of Internet routing with a simple and lightweight attestation mechanism. TBGP introduces a set of route update and withdrawal rules that, if correctly enforced by each router, can guarantee the authenticity and integrity of route information that is announced to other routers in the Internet. Through this, TBGP builds a transitive trust relationship among all routers on a routing path. We implement a prototype of TBGP to investigate its practicality. In our implementation, we use identity-based signature (IBS) and trusted computing (TC) techniques to further reduce the complexity of security operations. The performance study show that TBGP can achieve significantly better convergence performance and lower computation overhead than existing secure BGP solutions.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*

General Terms

Security, Design

Keywords

Routing, BGP, Hijacking, Secure BGP, Prevention

1. INTRODUCTION

The Border Gateway Protocol (BGP) is the only widely deployed inter-domain routing protocol connecting different IP networks or autonomous systems (ASes) to construct the whole Internet [24]. In ordinary BGP, every AS announces its route information with different prefixes. However, its neighboring ASes cannot validate this route information, but rather directly propagate it across the Internet. Obviously, this weak trust model allows forged route announcement propagations, which is a fundamental security weakness of BGP. Forged routes, which can be generated by configuration errors or malicious attacks, can cause large-scale network connectivity problems. For instance, on Feb. 24th, 2008, Pakistan Telecom (AS17557) started an unauthorized announcement of the prefix 208.65.153.0/24. One of Pakistan Telecom's upstream providers, PCCW Global (AS3491) forwarded this announcement to the rest of the Internet, resulting in the hijacking of YouTube traffic on a global scale [6]. The situation could be worse if forged routes are generated by remote attacks [10].

In order to effectively eliminate false announcements and improve the security of BGP, several security-enhanced BGP solutions have been proposed. They generally can be classified into two categories: *cryptography-based prevention* [21, 31, 30, 8, 18], and *anomaly detection* [28, 13, 19]. Cryptographic approaches, such as SBGP [21] and SoBGP [31], use a centralized routing registration authority and public key infrastructure (PKI) to ensure the authentication of routing announcements, These solutions are not sufficient to prevent data-plane attacks, where an AS can announce a route not adopted by itself [12]. Moreover, they usually consume a significant amount of extra router resources including computation and storage, and exacerbate the routing convergence performance. It is obvious that pure cryptography-based solutions are not cost-efficient to defend against routing attacks, and this impedes their deployment on the Internet. On the other hand, anomaly detection approaches aim to discover underlying hijacks in BGP announcements, e.g., by comparing BGP announcements with out-of-band information and querying third-party routing services [13]. However, most of the anomaly detection solutions raise false positives and require network operators to take actions in order to block detected anomalous routes [28, 13, 19].

In this paper, we propose a trusted BGP scheme called TBGP, which aims to use minimal computation cost to achieve BGP security goals. Unlike existing cryptography-based approaches, we do not solely rely on cryptography mechanisms to secure routing. Instead, we propose a set of well-defined route update and withdrawal rules that are enforced by the filters of each BGP router along a routing path. These rules guarantee that route announcements comply with the BGP specification [24]. Thus, the enforcement of these rules provides automatic route authenticity in each router and prevents the spread of forged routes over the Internet. In order to ensure that these rules are not misconfigured or maliciously modified, and hence correctly enforced on each router, TBGP introduces an attestation service running on each router. With this service interface, a neighbor router can challenge this router's current running

ASIACCS '11, March 22–24, 2011, Hong Kong, China.

state, including the integrity of its routing protocol stack, the routing rules, and the attestation service itself. When this attestation verification succeeds, the attesting router has the assurance that the route information it receives from this router is legitimate and follows the routing specification. Thus, the router can use the route information to update its own routing table, or announce it to its neighbors based on the same set of rules. In turn, its routing state and enforced rules can be challenged by other router. Thus, a *transitive trust relationship* can be built by attesting and verifying only one neighboring router along a routing path. TBGP exploits the transitive trusts among routers to extensively save computation and network resources compared to traditional secure BGP approaches.

The above attestation does not prevent a malicious router from claiming to own a particular AS number and generating forge routes. In order to verify the owner of an AS number and the authorization of using it, each route update is digitally signed by the attestation service upon the successful attestation challenge. A router is authorized to use its own private key to sign any valid announcement only when routes are successfully attested in OUT filters. The signature is then verified by its neighbors via their own attestation service. As the private key is bound with the AS number owned by the router, the attestation process can guarantee the authenticity of announced routes of a benign router.

We implement a prototype to demonstrate the practicality of TBGP, and use commodity techniques to further improve its performance. First, with the advent of Trusted Computing (TC) technologies, we note that TC-enabled chips are equipped in almost all commodity PCs and are ready for embedded systems [7, 15, 20]. Thus, we use this facility to securely store the private keys in each router, and bind the integrity of router software and the correct enforcement of BGP rules with authorized signing operations using the protected keys. Furthermore, we accomplish the verification of prefix originals and AS_PATH with the identity-based signature (IBS) scheme [9], which eliminates the centralized certificate management infrastructure and the aggregated signatures as in traditional RSA- and DSA-based algorithms. This significantly reduces the overhead of runtime security operations.

Our security analysis shows that TBGP achieves the security requirements of BGP, including AS number authentication, BGP speaker (router) authentication, AS path authentication, and prefix origin authentication. It also effectively prevents data-plane attacks such as traffic attraction attacks [12] by guaranteeing normal BGP execution routines and enforcing route attestation rules in each BGP speaker. In addition, we evaluate the performance of TBGP with both experimental studies and simulations. The experimental studies show that TBGP only introduces by an average of 2-ms delay in route selection and announcement of every route (per-prefix). We then seed the experimental data as the parameters into large scale simulations. Our simulation results show that TBGP has significantly lower performance overhead and resource consumption than existing secure BGP approaches. When compared to prior secure BGP solutions, TBGP has an improvement of at least 1.25 times in convergence time and 9.26 times in memory consumption. This evidently shows that TBGP could be a potential solution for building a trustworthy Internet routing infrastructure.

The remainder of the paper is organized as follows. In Section 2, we introduce the problem statement of BGP security and existing solutions, and the design goals of TBGP. In Section 3, we propose the BGP route rules and attestation algorithms, and present security analysis. The implementation details of our prototype is illustrated in Section 4. Section 5 presents performance evaluation results. Section 6 concludes this paper.

2. BACKGROUND AND DESIGN GOALS

2.1 BGP Security Threats

Current BGP is always under attacks from maliciously misconfigured speakers or intercepted unauthorized BGP sessions, both of which can cause BGP routing anomaly and further Internet disruption. Since BGP speakers fail to verify the origins of BGP announcements, a BGP speaker can announce any prefix that does not belong to its AS. Similarly, a BGP speaker cannot validate the AS path of a received BGP announcement. Thus, the announced route may be invalid and redirect traffic to wrong/malicious destinations. In general, there are two types of attacks in BGP: *prefix hijacks* and *invalid path attacks*.

Prefix hijacks include the *complete prefix* and *sub-prefix* hijacks. It is easy to carry out complete prefix hijacks on the Internet, but it is relatively hard to detect them. For example, a complete prefix hijack can occur when an AS announces itself as the origin of a prefix that it does not own, and its neighboring ASes then reroute any traffic with corresponding destination to the hijacker. The attack (1) shown in Figure 1 is a complete prefix hijack, in which a malicious speaker in AS 6 announces that AS 6 is the owner of the prefix 12.34.8.0/24 and advertises AS path {6} to AS 4. The sub-prefix hijack is similar to the complete prefix hijack except that its announced prefix is a subset of another announced prefix.

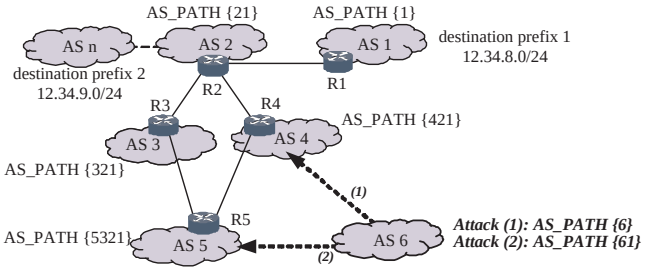

Figure 1: Examples of normal route and malicious route announcements of BGP. (1) indicates a complete prefix attack, and (2) an invalid path attack.

An invalid path attack occurs when the AS path of a BGP announcement includes fake AS numbers. For instance, the attack (2) illustrated in Figure 1 is an invalid path attack, where AS 6 advertises a forged AS path {61} to AS 5 and any traffic to AS 1 is redirected to AS 6 if AS 5 adopts this route. Because BGP is a policy vector routing protocol, it cannot detect relationships between ASes. Therefore, it is also very hard to detect invalid path attacks.

2.2 Related Work

Several security-enhanced BGP solutions have been proposed in literature, and they can be classified into two types in general. The first type uses cryptographic algorithms to provide the authentication of ASes and AS_PATHs (i.e., sequences of ASes that represent routes), such as SBGP [21], SoBGP [31], psBGP [30] and SPV [18]. The second type is to deploy invalid route detection mechanisms, such as IRV [13], Listen and Whisper [28], PGBGP [19], iSPY [32] and NetReview [16].

SBGP is the first proposed secure BGP solution [21], which uses public key infrastructure (PKI) to issue AS and prefix certificates for verification of announced prefixes and AS paths. In SBGP, aggregated signatures are used to guarantee the authenticity and integrity of BGP announcements. For a specific route, different signatures of prefix and AS_PATH are attached in announcements by traversed ASes. To improve the performance of SBGP,

S-A and SAS apply cryptographic operation speedup and sequential aggregate signature, respectively [33]. SPV adopts a more efficient cryptographic mechanism [18]. However, these solutions have the drawback of large computation and memory costs. To address these issues, Secure origin BGP (SoBGP) uses a distributed trust model [31], in which a new BGP message is introduced to deliver certificates. Unfortunately, SoBGP cannot prevent invalid AS path attacks. Pretty secure BGP (psBGP) uses a signed prefix assertion list (PAL) that consists of a number of bindings of AS numbers and (zero or more) IP prefixes [30]. Similar to SBGP and SoBGP, it is difficult to apply psBGP in real application scenarios where customer ASes may obtain IP addresses from different ISPs in a hierarchical way. Shi *et al.* [27] propose a memory attestation solution (BIND) to attest routes using trusted computing. However, this solution is quite time- and memory-consuming and may exacerbate BGP convergence performance, e.g., there are several communication rounds for encrypting/decrypting message authentication codes and attesting a received route update in a BGP speaker. In summary, these solutions usually need to consume large computation resources and cannot meet the practical requirements of real scenarios. Moreover, they usually cannot prevent data-plane attacks [12], because they cannot detect the inconsistency between BGP's control-plane (calculated routes) and data-plane (routes to forward packets).

Inter-domain route validation (IRV) [13] introduces an additional route validation service in BGP, through which the authenticity of BGP route information is verified. However, IRV cannot detect forged AS attacks. The Listen and Whisper solution [28] monitors all exchanged route announcements to detect underlying anomalies but offers weaker detection capability [33]. Moreover, Pretty Good BGP (PGBGP) [19] blocks large-scale attacks by delaying the propagation of suspicious routes. Recently, several improved prefix hijack approaches have been proposed. Lad *et al.* [22] propose an alert system to detect prefix hijacks by detecting the changes of prefix origins. Hu *et al.* [17] improve the detection accuracy by analyzing conflicts in data-plane footprints. Zhang *et al.* [32] propose iSPY to detect hijacks by analyzing prefix reachability in prefix owner networks. N-BGP [25] is proposed to build a trusted third party to realize a policy monitor using trusted computing (TC). N-BGP enforces route attestation rules for routing anomaly detection with a BGP monitor, but not in individual BGP speakers. These solutions can be easily deployed on the Internet without modifications to BGP and provide incremental approaches to secure BGP and are orthogonal to cryptography based secure BGP solutions.

Recently, Haeberlen *et al.* [16] propose NetReview to detect routing anomaly caused by attacks and misconfiguration using fault patterns and checking tamper-evident logs with these patterns in NetReview servers of ASes. In NetReview, routing messages are recorded in a tamper-evident log to analyze anomalous behaviors of BGP routes based on defined fault patterns. In this way, NetReview can detect invalid routes caused by attacks or configuration faults and policy conflicts. However, NetReview does not address the response mechanism to detected faults. Different from NetReview, which detects BGP faults based on fault patterns, TBGP enforces route attestation rules to guarantee normal behaviors of BGP routes. TBGP focuses on the prevention of forged routes caused by unintended or malicious misconfiguration, but does not address detection/prevention of policy conflicts, which we believe can be improved by configuration static analysis [11].

2.3 Design Goals of TBGP

From a security perspective, TBGP seeks to defend against different kinds of BGP attacks and guarantee the availability of BGP routes and normal packet forwarding in the presence of adversaries. We identify the following security goals [30] [1].

- *AS Number Authentication.* BGP speakers can verify whether an AS is the real owner of an AS number and is authorized to use the AS number.
- *BGP Speaker Authentication.* BGP speakers can verify whether a speaker is legal to announce prefixes, so as to guarantee that the BGP speaker is associated with an AS number.
- *AS Path Verification.* BGP speakers can verify whether the AS_PATH $\{AS_1, AS_2, ..., AS_n\}$ of a BGP route m for a prefix f_i is in the specified order. That is, m is generated from the prefix owner of AS_1, and has traversed $AS_2, \cdots, AS_n$.
- *Prefix Origin Authentication.* BGP speakers can verify whether an AS_n is authorized to generate an IP prefix f_i. In order to achieve that, one of the following three conditions should be verified: (1) The prefix f_i is indeed held by AS_n; or (2) AS_n is authorized to be the owner of f_i; or (3) AS_n is assigned by a set of prefixes F_i and has received another set of prefix F_j, such that f_j is aggregated from F_i, F_j, or both, and $\exists f_j \subseteq f_i$, where $f_j \subseteq F_i \cup F_j$.

Furthermore, in order for a secure BGP solution to be practically deployable on the Internet, the following goals should be satisfied.

- *Acceptable Performance.* A secure BGP solution should introduce minimal performance overhead (e.g., CPU cycles, memory footprint, and communication cost) over ordinary BGP, and does not significantly degrade the performance of a BGP speaker and the convergence performance of BGP.
- *Incremental Deployment.* A secure BGP solution should be partially deployable without disruption, which means that a subset of entities (e.g., routers, ASes, or ISPs) can deploy the solution without incurring loss of network connectivity.

3. DESIGN OF TBGP

For clarity, we initially assume that TBGP is fully deployed (i.e., on all participating routers in the network), and the allocation of AS numbers and IP prefixes to ASes is certified by authorities. We will discuss the solutions for efficient cryptographic operations and incremental deployment in later sections.

3.1 Overview

Ordinary BGP provides configurable filters called *IN filters* and *OUT filters*, which filter incoming and outgoing routes, respectively. With the filters, operators can configure their routers to discard routes that violate certain conditions. Filters are used by providers to ensure that they only accept or announce routes from/to their neighbors. If all providers perform this correctly, the network would be safe from attacks. However, many networks cannot filter violated routes effectively, due to the difficulty to infer the validity of routes from different ISPs. Basically, TBGP is designed to attest routes to check whether they comply with the BGP specification in filters and provide an automatic route filtering mechanism.

In TBGP, a BGP speaker signs a route if it complies with a set of route attestation rules in the OUT filters. By verifying the signatures in the IN filter, a neighboring router can easily know whether the route is valid in terms of BGP specification. With this mechanism, a *transitive trust relationship* can be built among the routers along a routing path. The root of this trust relies on the prefix own-

[1]Since the consistency between control- and data-plane is a basic BGP property according to the BGP specification [24], we do not explicitly specify it here.

ers that sign the route with prefix private keys. Each BGP speaker verifies, in its IN filter, the signature piggybacked in a received route update from its neighbor. A successful verification means that the route is attested by the neighbor and is authentic, and the route in Adj-RIB-IN will be updated. The BGP speaker selects the best route for the prefix. If the best route is changed, the BGP speaker announces the selected routes to its neighbors. Before that, the BGP speaker attests the route under propagation according to route attestation rules. A route is signed by the private key of the AS number only if it has been successfully attested, and thus neighbor routers can easily check whether the router is trusted and authenticated by verifying the signature.

To illustrate the idea of TBGP, we refer again to Figure 1. Suppose AS 1 announces that it is the owner of prefix 12.34.8.0/24. Then R1 is authorized to send the AS PATH of the route {1} signed with its private key. R2 in AS 2 receives the route update and updates it in Adj-RIB-IN for route selection only if it successfully verifies the signature in the IN filter. If the route is selected as the best route to the destination 12.34.8.0/24 in R2, then R2 checks whether the route under propagation complies with the attestation rule. The route is authenticated only if the route is successfully attested. In this example, the AS_PATH of route under propagation is {21}, which prolongs the AS_PATH in the previously received route update. Then, AS 1 and AS 2 build trust between themselves. R2signs the AS PATH using its private key that correspond to the AS number. Similarly, R3, R4, and R5 verify the route in their IN filters and announce the route to their ASes with the correct signature. Thus, AS 1, AS 2, AS 3, AS 4, and AS 5 build a trust relationship for prefix 12.34.8.0/24.

Now, the routers in AS 6 cannot launch the prefix hijack attack (see Section 2) by announcing the ownership of the prefix 12.34.8.0/24 because they do not have the correct private keys to sign the routes for the prefix. Similarly, it cannot launch the invalid path attack (see Section 2) by propagating the forged route {61} because the route cannot be successfully attested by AS 5 (assuming that no router is compromised). In Section 4, we will discuss how to prevent forged routes if some routers are compromised.

Thus, TBGP effectively eliminates aggregate signatures of a full AS path in route attestations as in existing cryptography-based secure BGP solutions. The next two subsections explain more details of the route attestation rules and algorithms to build the transitive trust relationships between different ASes/routers.

3.2 Route Attestation Rules for TBGP

The trust of a BGP system depends on the expected behavior of each router when selecting and announcing route information. A set of route attestation rules is defined in TBGP, which, if correctly enforced by a router system, can guarantee the authenticity and correctness of its announced information.

For simplicity, in this paper we only consider the attestation rules for eBGP sessions, where we assume an AS only has one BGP speaker. The OUT filter of a BGP speaker checks whether an announced route follows the route attestation rules based on the information in the IN filter. The announcement is signed and further propagated only when it passes the check. A neighboring BGP speaker, upon receiving the announcement, first verifies if it is actually sent by a speaker that owns the AS number. If attestation verification succeeds, then it means the route is trusted, and the announcement is accepted. Thus, these two BGP speakers can build a trust relationship. This happens recursively along an AS_PATH. Thus, *there is no need for a BGP speaker to check and verify every hop in the AS_PATH*, i.e., prefix verification and AS_PATH verification for all speakers in the path. A neighboring BGP speaker only needs to verify limited information, such as the signature of prefixes or AS but not both. These attestation operations are enforced by a BGP attestation service (see Section 4). Through the built trust relationship, aggregated signatures are eliminated. Before we introduce the detailed rules, Table 1 gives the symbols used in these rules.

Table 1: Symbols used in route attestation rules

f_i, AS_n	IP prefixes, AS number
$AS[f_i]$, $AS(f_i)$	A set of AS_PATH for prefix f_i, a specific AS_PATH
$\Downarrow AS[f_i]$	AS_PATH in a received update for f_i
$\Uparrow AS[f_i]$	AS_PATH in the update for f_i under propagation
$Withdraw(f_i)$	A received withdrawal to prefix f_i
$PreList(AS_n)$	Prefix list owned or received by AS_n

DEFINITION 1. BGP Route Announcement Rule: *A BGP speaker is authorized to send a valid BGP announcement,* $Update(f_i, AS(f_i))$*, if and only if one of the following three conditions is true:*

- $f_i \subseteq PreList(AS_n) \wedge (\Downarrow AS[f_i] == \emptyset) \wedge (\Uparrow AS[f_i] == \{AS_n\})$*;*
- $((\{AS_n\}^+ \cup \Downarrow AS[f_i]) == \Uparrow AS[f_i]) \vee (\Uparrow AS[f_j] \subseteq (\{AS_n\}^+ \cup \Downarrow AS[f_i]) \wedge f_i \subseteq f_j)$*;*
- $(Withdraw(f_i) \vee AS_n \in AS[f_i]) \wedge ((\{AS_n\}^+ \cup AS(f_i)) == \Uparrow AS[f_i])$.

This rule illustrates that an announcement is valid if and only if (i) f_i is the owner of AS_n, (ii) or it is a re-announcement after a previous announcement, or (iii) it is an announcement after a previous announcement that does not include valid routes. We note that since a route update triggered by ISP policy changes is similar to that specified by the third condition of this rule, we do not discuss it explicitly. Note that this security rule considers the address aggregation and legal route prepending issues during route propagation. $\{AS_n\}^+$ in this rule denotes that it is legal to prepend its own AS number in an AS path.

The first condition in this rule describes that the advertisement speaker in AS_n is authorized to announce the prefix if it is the owner of the prefix, and the announced route should only contain itself in the AS Path. For example, AS 1 in Figure 1 is allowed to advertise AS path {1} to its neighbors. The second condition describes that the BGP speaker is allowed to advertise a route if it is a re-advertisement of a previous route and prolongs the AS path with its AS number, or the AS path in the re-advertisement route is a subset of the full AS path which is prolonged by including its AS number[2]. For instance, in Figure 1, AS 2 advertises the AS path {21}, which is legal if the AS path in the previously received route update from AS 1 is {1}. Suppose that AS 3 receives the AS path {21} for the destination 12.34.8.0/24 and receives the AS path $\{2n\}$ (for some AS number n) for prefix 12.34.9.0/24 in the route from AS 2. The announced route whose AS path is {21} for prefix 12.34.0.0/20 is allowed because it is the intersection of these two prefixes, and thus it is a legal route aggregation based on the second condition.

The third condition describes the situation that the announced route is legal if the route under propagation is the union of a record in previous received route updates after receiving a route withdrawal and a route announcement whose AS path including itself. For example, assuming that the link between AS 2 and AS 3 in Figure 1 fails, AS 3 then withdraws the route to AS 5. Since AS 5 has

[2]Actually, route disaggregation is similar to the route aggregation. In general, AS should achieve another type of secret keys different from the prefix owner keys if it announces itself as the origin of the aggregated/disaggregated prefix. However, this process is application-specific, and we do not discuss it in this paper.

received a route update with AS_PATH {421}, which is recorded in the attestation service, the route attestation rule allows AS 5 to advertise the route with AS_PATH {5421} to its neighbor ASes. If AS 5 advertises a route whose AS path is not recorded, then the route under propagation is regarded as a forged one and dropped. In addition, if a BGP speaker receives a route containing its own AS number, e.g., the route oscillation cases discussed in [14], then it announces another recorded route, which is similar to the route withdrawal case above.

DEFINITION 2. BGP Route Withdrawal Rule: *A BGP speaker is authorized to send a valid BGP withdrawal, $Withdraw(f_i)$, if and only if the following condition is true:*

- $(Withdraw(f_i) \wedge AS[f_i] == \emptyset) \vee f_i \in PreList(AS_n)$.

Similarly, this rule describes that a route withdrawal is allowed if and only if f_i is the owner of AS_n or there is no available route record for prefix f_i in the attestation service. For example, assuming that the link between AS 1 and AS 2 fails in Figure 1, AS 2 does not have an available route to AS 1. Then, the BGP speaker in AS 2 is allowed to send route withdrawals to AS 3 and AS 4.

3.3 Trust Establishment

The above route attestation rules guarantee the validation of BGP announcements if they are really enforced on each router. We give the detailed algorithms to verify this via attestation service in the IN and OUT filters of a BGP speaker. As aforementioned, when a BGP speaker in AS 1 receives an announcement, it is firstly checked and verified by the attestation service in the IN filter. Algorithm 1 shows the algorithm of the attestation service in the IN filter. If the received announcement is sent by the owner of a prefix, then the prefix string will be used to verify the signature. As shown in Figure 2, through verification, the identity of the originating BGP speaker in AS 1 and the ownership of the prefix are validated in AS 2. This is the first level of a trust relationship for prefix 12.34.8.0/24. If the announcement is propagated to AS 3 by a delegated BGP speaker in AS 2, then we need to verify whether the speaker of AS 2 is authorized to propagate this route. Thus, the AS number of AS 2 is used to verify whether the BGP speaker is an authentic owner of AS 2. If the announcement is verified in the IN filter of AS 3, then then AS 3 can trust the announcement because the successful verification means that the received AS_PATH is composed with previous consecutive trusted ASes. Thus, the received route should be updated as an active record and stored in the route database for further attestation by the OUT filter. Similarly, AS 4 can build trust with AS 2 by verifying the announcement.

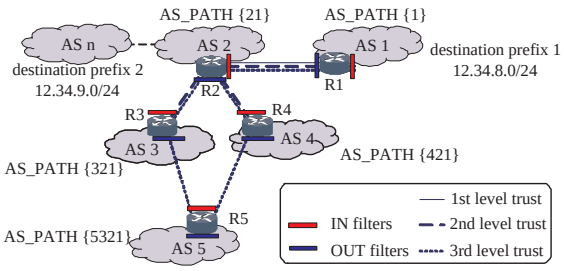

Figure 2: Building transitive trust between ASes/routers.

After a BGP speaker completes a route selection process, the chosen route is further propagated if there is a change in the route. The route announcement will be checked in the OUT filter. Algorithm 2 shows the attestation algorithm in the OUT filter. First, the active record *R* of the route record updated in the IN filter, which

Algorithm 1 Attestation in BGP IN filters.

Input: prefix f_i;
AS_PATH [AS_{n-1}, ... , AS_1];
signatures [$sig_{AS_{n-1}}${AS_{n-1}, ... , AS_1}, sig_{f_i}{f_i}];
Output: **true**: the announcement will be proceed;
false: it will be dropped;
1: **if** length(AS_PATH) == 1 **then**
2: verify(f_i, [$sig_{sQ_{ID_{f_i}}}${f_i}]);
3: **if** signature is invalid **then**
4: return **false**;
5: **end if**
6: **else if** length(AS_PATH) $>$ 1 **then**
7: verify (As_{n-1}, [$sig_{sQ_{ID_{AS_{n-1}}}}${AS_{n-1}, ... , AS_1}]);
8: **if** signature is invalid **then**
9: return **false**;
10: **end if**
11: **end if**
12: update_record(f_i, [AS_{n-1}, ... , AS_1]);
13: **if** exist_db(f_i, AS_{n-1}) **then**
14: update_db(f_i, [AS_{n-1}, ... , AS_1]);
15: **else**
16: insert_db(f_i, [AS_{n-1}, ... , AS_1]);
17: **end if**
18: return **true**;

triggers the route re-computation, is located. If the record does not exist or does not match a received route, which means that the route is sent by the owner or the route update follows a received route not including a valid AS path, then all the records of the prefix are fetched from the route database. Then the attestation service checks whether the announcement is allowed based on the route attestation rules. If the announcement is legal, then it will be signed and sent to the neighboring speaker. The signature is either based on the private key of f_i (i.e., $sQ_{ID_{f_i}}$) if the AS is the owner of f_i, or based on the private key of AS_n (i.e., $sQ_{ID_{AS_n}}$) otherwise. As Figure 2 shows, after AS 3 and AS 4 successfully attest the route in OUT filters, the trust relationship will be extended to AS 5 if it adopts the route.

Algorithm 2 Attestation in BGP OUT filters.

Input: prefix f_i;
AS_PATH [AS_{n-1}, ... , AS_1];
Output: **true**: the announcement will be sent out;
false: it will be dropped;
1: **if** ($R \Leftarrow$ locate(f_i, AS_{n-1})) == **null then**
2: $R \Leftarrow$ locate_db(f_i);
3: **end if**
4: **if** check(f_i, R, [AS_n, AS_{n-1}, ... , AS_1]) == **false then**
5: return **false**;
6: **end if**
7: **if** f_i is owned **then**
8: $S \Leftarrow$ sign($sQ_{ID_{f_i}}$, {f_i});
9: **else**
10: $S \Leftarrow$ sign($sQ_{ID_{AS_n}}$, [AS_n, AS_{n-1}, ... , AS_1]);
11: **end if**
12: return **true**;

With these algorithms, the validity of BGP announcements is guaranteed by enforcing verifications and route attestations in the IN/OUT filters of BGP speakers. That is, the identity of a BGP speaker is verified and its route authenticity is guaranteed by the route attestation rules, through which different speakers build a transitive trust relationship. Specifically, when a BGP speaker in AS_y receives a route update of a prefix with the correct signature from a speaker in AS_x (i.e., the route update is attested by AS_x

itself), it attests the route update by verifying the signature and puts this route in Adj-RIB-IN for future route selection. Thus, AS_x and AS_y build trust between them for this prefix. Similarly, if the route for this prefix is adopted and further propagated to AS_y's neighbors $\{AS_k,...AS_n\}$, the update is attested in the OUT filter of AS_y speaker and then is signed by its private key. Thus, all ASes can build trust with each other, and the trust relationship is transitive by signing/verifying signatures and enforcing the rules in the IN and OUT filters. For example, as shown in Figure 2, the second level of the trust relationship among ASes 1, 2 and 3 is built if AS 1 attests the route in the OUT filter and ASes 2 and 3 successfully verify the route in their IN filter. Similarly, the third level trust relationship is built among ASes 1, 2, 3, 4 and 5 if AS 3 and 4 attest the route in their OUT filters and AS 5 verify it in its IN filter. Any forged routes cannot be successfully attested by the attestation service. That is, an AS can trust routes from neighbor ASes if and only if the routes are verified, which means that the routes are strictly attested by neighbor ASes themselves. Therefore, TBGP can effectively defend against forged BGP routes no matter whether they are generated by configuration errors or malicious attacks. Each AS only needs to attest route updates with the keys of the last hop and does not need to attest them with every hop information. We can prove that TBGP achieves the following four security goals: AS number authentication, BGP speaker authentication, AS path authentication, and prefix origin authentication. In the interest of space, we refer readers to [23].

4. PROTOTYPE IMPLEMENTATION

We implement a prototype of TBGP and demonstrate its practices. Our prototype solves the following two questions which are important for real deployment on the Internet:

1) How to reduce the complexity of cryptographic operations in TBGP? Is it possible to eliminate distribution and management of thousands of public keys in traditional secure BGP proposals?

2) How to realize a tamper-resistant TBGP such that it can guarantee the integrity execution of route attestation algorithms and rules, and thus preserve the consistency of routing control- and data-plane? In other words, can we ensure that a TBGP router cannot pretend to be a trusted one if the system is compromised, e.g., the route attestation service is disabled or routing control- and data-plane are not consistent?

Our TBGP solution is built on two existing key techniques: identity-based signature (IBS) and trusted computing (TC). In this section, we present three primitive functions used in TBGP based on these two techniques: (i) secure storage of BGP keys, (ii) signing/verifying BGP announcements, and (iii) BGP attestation service.

4.1 Preliminaries

Identity-Based Signature Algorithm Identity-based cryptography (IBC), which is an alternative to the traditional certificate-based public key cryptography, uses user identity information (e.g., email address) as the public key [26]. The private key in an IBC system is generated by a private key generator (PKG) according to the user identity information. IBC is firstly designed by Shamir and resolves the problem of key storage and management in certificate-based cryptographic algorithms. IBC includes identity-based encryption (IBE) and identity-based signature (IBS) algorithms [26]. In our implementation of TBGP, we use IBS to verify and validate announced prefix and AS_PATH. Specifically, an IBS system consists of four basic algorithms: *Setup* algorithm generates a set of public system parameters and private master secret; *Extract* algorithm extracts the private key corresponding to a given public key, which takes the system parameter, the master secret, and the public key (a public ID) as inputs; *Sign* algorithm returns the signature of a given message using the system parameters, a private key, and the message as inputs; *Verify* algorithm uses the system parameters and an ID to check whether a signature is valid, i.e., the message is signed with the corresponding private key and is not altered. With IBS, TBGP routers do not need to obtain different public keys before route attestation in advance. Thus, TBGP eliminates the centralized certificate distribution and storage, and reduces the complexity of security operations.

As an example, suppose that Alice wants to send a message to Bob who wants to verify the message signature. First, both Alice and Bob need to retrieve the system parameters and their respective private keys, which are generated by a private key generator (PKG) with the *Setup* and *Extract* algorithms, along with their respective public IDs. After this, Alice then signs the message with the *Sign* algorithm. After receiving the message from Alice, Bob does not need to retrieve Alice's public key, which usually takes place in conventional certificate-based public key algorithms. Instead, Bob simply verifies the message with Alice's ID and checks whether the signature is valid with the *Verify* algorithm.

Trusted Computing The Trusted Computing Group (TCG) [5] has defined a set of hardware and software specifications for Trusted Computing (TC) technologies. The root-of-trust of the TCG architecture is the Trusted Platform Module (TPM), a discrete chip which performs certain cryptographic functions and provides secure storage. TPM provides secure storage for high level applications and services, which is leveraged by TBGP to protect IBS private keys and guarantees that a signature can only be generated when a BGP routine is correctly executed and route attestation rules (cf. Section 3.2) are enforced without disabled or maliciously modified. Specifically, a router receives a private key from a PKG and seals (encrypts) it with a key protected by its TPM when it joins the Internet. When generating a signature, the TPM unseals (decrypts) this key only when certain configurations of the system can be identified, which are represented by Platform Configuration Registers (PCRs) inside the TPM. Through this mechanism, the private key is always protected, the resulting signature is guaranteed to be signed by the proper private key, and the signature is signed only under known good platform state, e.g., the integrity of the attestation service and rules is maintained.

Remote attestation is another important TC mechanism used by TBGP. When a router initially joins the Internet, in order to get permissions to announce routes, it needs to get its private keys. For this purpose, its platform should be attested by the authorities before the router provides its routing service. The TPM on the router signs the value of system state and sends it to an authority, which verifies if the current platform is in a good state. Upon successful verification, the authority releases corresponding private keys to the router, which in turn seals them with TPM. This guarantees that a private secret is only released to a good router. Once private keys are achieved in a router, TPM protects the keys locally. Combined with the secure storage mechanism above, a protected key is only available for signing when the system is in the same good state as when the key is retrieved and installed. Thus, it lays the foundation for trust establishment between BGP speakers, which is the prerequisite to ensure that route attestation rules are enforced in TBGP.

4.2 Primitive Functions of TBGP

TBGP leverages three core mechanisms to achieve the security goals: secure storage of BGP keys, signing/verifying BGP announcements, and BGP attestation service. These mechanisms jointly provide the functions of route attestation. Before introducing the details, we assume that BGP speakers in TBGP are equipped with

TCG-compatible TPM chips for key storage and the attestation of the BGP process and route attestation rules. Several designs of TPM for embedded systems have been proposed [7, 5]. Alternatively, secure software TPM (swTPM) [4], a kernel module in the router OS, can be used if hardware TPM is not available.

Secure Storage of BGP Keys The secure storage mechanism in TBGP is realized by directly applying the secure storage primitive provided by TPM. In TBGP, all sealed keys can be unsealed from TPM and used by the BGP attestation service only when the BGP system running on a router is not maliciously changed. In general, TPM in a BGP speaker seals private keys sQ_{ID}, which includes $sQ_{ID_{f_i}}$ corresponding to its owned prefixes, and $sQ_{ID_{AS_n}}$ corresponding to AS number AS_n. In TBGP, similar to traditional BGP security solutions [21, 30, 18], we also assume some trusted address assignment authorities, such as ICANN and IANA, and other trusted delegation organizations act as PKGs to generate and distribute private keys and public parameters to routers before they are deployed on the Internet. Note that, for the strong security purpose, address assignment authorities should collaborate with router vendors who provide fingerprints of different BGP softwares with route attestation rules to accurately attest BGP systems before assigning private keys. Once a router obtains its private keys, all keys are sealed into the TPM. Above three steps in setup stage are illustrated in Figure 3.

When a BGP router is in a good state, all the keys can be unsealed for later signing operations. The good state means that the values represent the expected software runtime of the router, e.g., identical to the values when the keys are sealed. That is, the BGP system is not compromised and the security configurations of TBGP are not maliciously changed. Thus, we have the assurance that: 1) announced routes to neighbors are identified to be used for forwarding packets, which guarantee the consistency of control and data planes; 2) the route attestation rules of TBGP are well enforced during the runtime of a BGP system and are not changed/disabled by its operators. All these are checked during router bootstrapping (cf. Section 4.2). To preserve a good runtime environment, several runtime protection mechanisms can be used, such as ARM TrustZone, Intel's Trusted Execution Technology and AMD's Pacifica technology [15], which are out of the scope of this paper.

Signing/Verifying BGP Updates In TBGP, all outgoing BGP updates (i.e., the routes that a router propagates to others) need to be signed by the router, and all incoming BGP updates (i.e., the routes that a router receives from others) need to be verified by the router before adopting them. The signing/verifying operations include the prefixes and AS_PATH of an announcement. Figure 3 shows the work flow of these operations with IBS. After obtaining the keys and system parameters of IBS, a BGP speaker A signs an announced route using its keys associated with its owned prefix (if the prefix is owned) or its AS number (if the prefix is not owned), and a neighbor speaker B verifies the received announcement using the corresponding public key of speaker A (e.g., the ID string corresponding to the prefix keys or AS keys in the signing procedure). Speaker B can easily determine which string to use to verify the announcement because the prefix and AS public keys are denoted in the BGP update. For example, if speaker B receives a prefix announcement from speaker A, then it uses the AS number ID of speaker A to verify the signature of the announcement. Thus, the public key distribution and management problem in PKI-based BGP schemes is well eliminated in TBGP. If the signature verification fails, speaker B drops the announcement. As aforementioned, a successful signature verification by speaker B implies that the announcement is signed with speaker A's appropriate private key within a good BGP runtime system, i.e., the route attestation rules are correctly enforced by speaker A. To prevent route replay attacks, speaker A also signs route announcement with a timestamp.

Notation:
ID_{AS}: AS number
MK: The master key of PKG
Q_t: A string generated and kept in PKG
params: The parameters known to all TBGP routers
sQ_{ID}: The private key corresponding to ID
SK(m): The ciphtertext of message m

IBS setup at PKGs:
1) PKGs: (Q_t, sQ_{ID}) = Keygen(*ID, MK, params*);
2) PKGs -> Routers: SK(sQ_{ID}, *params*);
3) Routers -> TPMs: TPM_Seal(sQ_{ID});

Route Update:
4) Router A: f= $\text{Sign}_{\{params,\ sQID\ \}}$(Announcement);
5) Router A-> Router B: (Updates | f);
6) Router B: $\text{Verify}_{\{params\ |\ ID\}}$($f$).

Figure 3: The IBS procedure in TBGP.

BGP Attestation Service The attestation service in TBGP provides interfaces for verifying and attesting BGP updates by a BGP speaker, and provides the mechanism to verify if route attestation rules are enforced by the speaker. Through this, transitive trust relationships can be built between BGP speakers. Basically, there are three major interfaces for BGP speakers: service initialization, validation in the BGP ingress filter (IN filter), and validation in the BGP egress filter (OUT filter). The interfaces are shown in Figure 4 and the attestation algorithms are discussed in Section 3.

The BGP attestation service initialization is invoked by a router system during its bootstrap phase after the integrity of the BGP system, including the BGP software and the route attestation rules, are validated by the trusted components on the platform built upon TPM. This interface requires two parameters: the hash values of BGP routing system and the route attestation rules. Note that different routers from different router vendors have different BGP system releases and thus different hash values. If these two parameters are not tampered, then the routing system can be launched successfully. Otherwise, it quits directly. After the BGP system is launched, all these parameters are reported into PCRs of its TPM. After this, the BGP system and attestation service can use private keys sealed by the TPM. The procedure is discussed in Section 4.1. If the attestation service is disabled, the BGP system cannot achieve the private keys and thus is unable to sign any route update. We will demonstrate this in Section 4.3.

Interface of BGP Attestation Service
0. *Initialization*: evaluated by core root of trust
   in BGP systems including programs and security rules,
   out launch BGP system and load rules and store into PCRs;
1. *IN filter*: incoming BGP updates
   in BGP updates (prefix | AS_PATH),
   in BGP attributes in updates including update signature,
   out true or false? // accept or reject BGP update;
2. *OUT filter*: outgoing BGP updates
   in BGP updates (prefix | AS_PATH),
   out true or false? // continue or drop BGP update,
   // *if update legal* sign prefix and AS_PATH,
   // *else* drop.

Figure 4: The interfaces of BGP attestation service on a router.

The IN filter and OUT filter interfaces in TBGP are placed in the same places as those in existing BGP protocol on a router [24]; that

is, they are invoked after receiving BGP updates and before sending BGP updates, respectively. When a speaker receives a BGP update, its attestation service verifies and validates the prefix string or AS number in the announcement in the IN filter of BGP protocol. If the verification fails, the announcement is dropped; If the verification succeeds, the attestation service will record the route information for later route attestation[3]. After BGP route selection process completes, the speaker may announce updated routes to neighbors. In the OUT filter, the attestation service is invoked again, which first locates the recorded route information corresponding to routing re-computation, and checks whether the announced routes comply with route attestation rules together with the located information. The outgoing routes are dropped when they do not comply with the route attestation rules, e.g., they are tampered by network operators.

4.3 Prototype Implementation

We implemented the TBGP in Zebra BGP daemon [1] with software TPM [4]. We use the IBS implementation in MIRACL cryptographic library from Shamus Software [2]. Our prototype implements three primitive functions described above using less than 3,000 lines of C codes.

In TBGP, if the BGP process is tampered, it cannot achieve the private keys from TPM, although it still can be booted and executed. This ensures that all route updates cannot be signed no matter whether they comply with attestation rules or not. If key unsealing succeeds, the BGP attestation service obtains private keys and attests route updates received (sent) from (to) neighbors in the IN (OUT) filter. The route updates are also signed and verified in IN and OUT filters if they are successfully attested.

In many existing BGP solutions, data-plane attacks [12] can be launched by modifying the records in Adj-RIBs-OUT and hence making the records in Local RIB and Adj-RIBs-IN inconsistent. Since the BGP process is attested with TPM and the consistency between a router's control-plane and data-plane can be attested and verified by its neighbors, any tampered BGP process whose records in the control- and data- plane are not consistent will be unable to announce routes with correct signatures, and hence the routes announced by them will not be adopted by their neighbors. Thus, data-plane attacks can be prevented in TBGP.

5. PERFORMANCE EVALUATION

We use both experiments and simulations to evaluate the performance of TBGP. The simulated and experimental networks have one BGP speaker for each AS. We believe it is reasonable and sufficient for performance study because most secure BGP schemes focus on inter-AS communications.

For our experiments, we deploy our TBGP prototype in three Linux-2.6.21 machines which have Pentium 4 1.7GHz CPU and 1GB memory and form a topology of 3 ASes shown in Figure 5. ASes 1 and 2 have a peering link, and ASes 2 and 3 have a peering link. We only configure different number of prefixes in AS 1, and AS 2 only forwards the learned route to AS 3. We study the overhead of different operations in TBGP: 1) *update sending operations*: the duration between the time when routes are selected as the best route and the time they are sent out to neighbors; 2) *update reception operations*: the duration between the time when route updates are received and the time they are selected as the best route; 3) *entire operations*: the duration between the time when route updates are received and the time they are sent out to neighbors. We evaluate the overhead in update sending in AS 1, the overhead in overall update process in AS 2, and the overhead in update reception in AS 3.

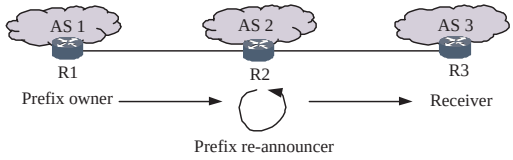

Figure 5: AS topology in our experiments. AS 1 announces prefixes, and AS 2 forward the routes to AS 3.

We further simulate TBGP to study its performance in large scale networks. Similar to most of the previous BGP proposals (e.g., [33]), we use SSFNet [3], which is an event-driven simulator, and provides basic process model of BGP [3]. The experimental performance is seeded into simulations as the parameters. We use four different scales of AS-level topologies with 10, 29, 110, and 208 ASes, respectively (the later three topologies provided by BJ Premore [3] are generated from real BGP routing tables and used in most of BGP simulations [33, 29]). In our simulations, we compare TBGP with different variants of SBGP schemes, ordinary SBGP, SBGP with cryptographic operation speedup (S-A) [33], and SBGP with sequential aggregate signature (SAS) [33]. The performance of cryptographic operations in these existing schemes is measured with standard Digital Signature Algorithm(DSA) [33]. Since SPV improves the convergence performance with the cost of memory consumption for cryptographic key storage and is similar to S-A [33], we do not evaluate it in our simulation.

5.1 Experimental Data

Firstly, we evaluate the overhead introduced by key unsealing during BGP bootstrapping. Figure 6 shows that TBGP has about 33% delay in bootstrapping. Since it is only one-time operation, the overhead is acceptable. Furthermore, we evaluate the performance of 512 bits IBS algorithms in TBGP. The execution time of signing and verifying operation with IBS is about 4ms and 50ms, respectively. The overall overhead is similar to that of the RSA and DSA algorithms [2].

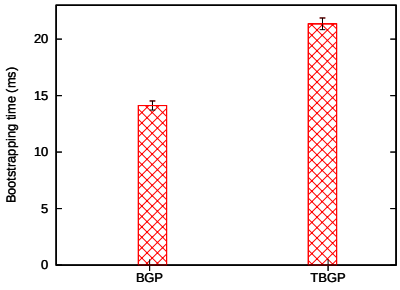

Figure 6: TBGP only introduces 33% delay in BGP bootstrapping, and the overhead is acceptable because bootstrapping is an one-time operation and does not impact the performance of routing selections.

The processing overhead in TBGP is introduced by route attestations. We evaluate the processing overhead of TBGP with different number of announced prefixes. All overheads increase with the increases of the number of announced prefixes. If we only announce 1 prefix in AS 1, the overall process time in ordinary BGP is 0.2 *ms*, and the overheads in update sending, update reception and entire operations are 0.009 *ms*, 2.2 *ms*, and 2.4 *ms*, respectively; if

[3]In our prototype, we directly leverage Adj-RIBs-IN to realize the database since it is tamper-resistant in our prototype.

we announce 100 prefixes in AS 1, the overall process time per prefix in ordinary BGP is 0.16 *ms*, and the overheads in update sending, update reception and entire operations per prefix in TBGP are about 2.1 *ms*. Figure 7 shows the ratio of processing overhead in TBGP over ordinary BGP. TBGP introduces 6% overhead in update reception, 35 times overhead in update sending, and 11 times overhead in entire operations in 1 announced prefix. It introduces about 13 times overhead in three different operations with other different number of announced prefixes. In the next subsection, we will study whether the processing overhead affects the performance of BGP routing (c.f., Figure 8).

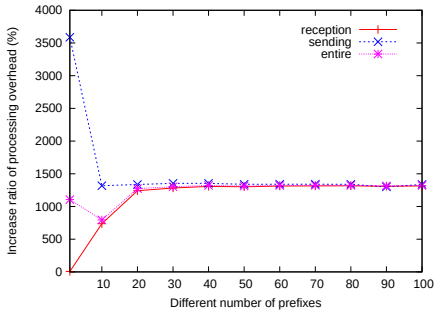

Figure 7: The overhead ratio of different operations between ordinary BGP and TBGP: update sending operations, update reception operations, and entire operations.

5.2 Simulation Results

It is not surprising that TBGP introduces communication and processing overheads compared to ordinary BGP, as it consumes CPU resources to perform IBS signing and verifying operations, which are the major causes influencing the BGP convergence performance. To explore these aspects, we simulate with 512 bits IBS algorithms and model running times in Section 5.1. We evaluate the routing convergence time of our simulation, which considers all the overheads introduced in TBGP route computation and selection, and is frequently used to evaluate computation overheads in literature. Figure 8 shows the impact of TBGP on convergence time, compared with the ordinary BGP. In these four different topologies, TBGP has 7%, 10%, 4%, and 0% extra convergence time compared to ordinary BGP, respectively. Especially, TBGP does not introduce extra convergence delay in large-scale topologies, such as the 208 ASes topology, because the MRAI timer [24] of 30 seconds becomes the major cause of convergence delay. Compared with SBGP, whose convergence time is over 200% larger than that in ordinary BGP [33], TBGP achieves much better performance.

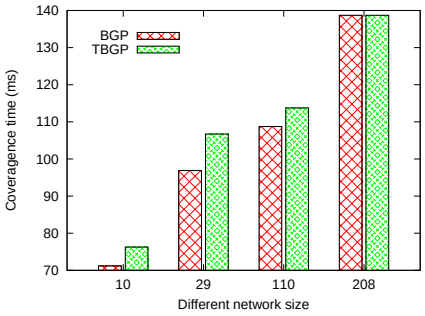

Figure 8: TBGP introduces average 5% of extra convergence time over ordinary BGP. Compared to 200% extra convergence time of SBGP, it introduces very small convergence overhead.

Figure 9 shows the impact of TBGP on the increase ratio of convergence time with the 110 ASes topology. TBGP only increases 4% convergence delay and achieves much better routing performance over SBGP and other variants of SBGP. For instance, the convergence performance in SBGP increases over 2 times of convergence delay, S-A introduces 9% extra convergence delay, and SAS increases over 3 times at the cost of increased memory consumption. Compared to SBGP, S-A and SAS, TBGP has 56.5, 1.25 and 75 times improvements in convergence time, respectively. The performance result is rational because only one signing and verifying operation is involved in a BGP speaker to attest a route in TBGP, while other secure BGP schemes need several times to verify a route. The overhead of message signature in TBGP is reduced from $\mathcal{O}(n)$ in SBGP to $\mathcal{O}(1)$ where n is the length of an AS_PATH. Note that to verify a received route update in these schemes, the time of signature verification is super-linear to the length of AS_PATH.

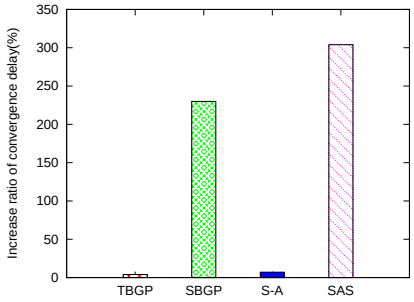

Figure 9: TBGP only has 4% increase ratio in convergence time relative to ordinary BGP in the 100 ASes topology. Compared to SBGP, S-A, SAS, TBGP has 56.5, 1.25, and 75 times improvements, respectively.

Figure 10 shows the impact of TBGP on message size and memory costs with 110 ASes topology. The baseline of average announcement message and memory cost in our experiment is 36.09 bytes and 9 KB [33], respectively. On average, the message size increase in SBGP is more than 763% and that in TBGP is only about 96%. Compared to S-A and SAS, TBGP still achieves much better performance. For example, the message size of BGP updates in S-A and SAS is 19.69 and 9.82 times larger than that in TBGP, respectively. Furthermore, TBGP has significant improvement in memory consumption. As illustrated, the SBGP scheme consumes additional 1140% of memory to cache routes and their signatures, but TBGP only requires about 1.1 times more memory to cache routes and has a 9.26 times improvement over sBGP. Similarly, memory consumption in S-A and SAS is about 130% larger than that in TBGP. The reason behind the low cost is that TBGP does not require caching received route signatures for further propagation thus eliminates the storage complexity of $\mathcal{O}(n^2)$ in SBGP.

6. CONCLUSION

In this paper, we propose TBGP, a lightweight secure BGP solution to prevent BGP routing attacks. In TBGP, route attestation algorithms are proposed to simplify route attestations and build a trusted Internet routing infrastructure. With these algorithms, a set of route attestation rules is strictly enforced in each router and thus aggregated signatures are eliminated without sacrificing the security of BGP. Our prototype leverages the trusted computing (TC) technology to build transitive trust relationships between BGP speakers, and the identity-based signature (IBS) algorithm to

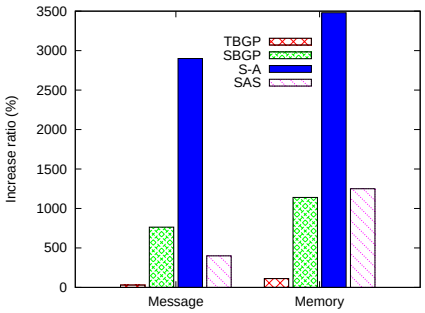

Figure 10: Overheads introduced by TBGP are much lower than other schemes. TBGP introduces 96% of increase in update message size and requires about 1.1 times more memory to cache routes. Compared to SBGP, S-A and SAS, TBGP has 24.38, 95.41 and 12.28 times improvements in message size, and 9.26, 30.32 and 10.25 times improvements in memory consumption, respectively.

sign/verify BGP routes and reduce the complexity of security operations in existing secure BGP solutions. Our performance study shows that TBGP has significantly better convergence performance and lower resource cost than traditional solutions.

Acknowledgements

We would like to thank Andreas Haeberlen, Lixia Zhang, Sharon Goldberg, Dah Ming Chiu, John C.S. Lui, and Jennifer Rexford for very helpful feedback and suggestions. This work is supported by NSFC grant No. 61073166, 973 Program grant No. 2009CB320502, 863 Program grant No. 2009AA01Z251, and the National Science & Technology Pillar Program of China, grant No. 2008BAH37B03.

7. REFERENCES

[1] GNU Zebra. http://http://www.zebra.org/.

[2] Shamus software ltd, MIRACL. http://www.shamus.ie/.

[3] SSF network models (SSFNet). http://www.ssfnet.org/homePage.html.

[4] TPM emulator. http://tpm-emulator.berlios.de.

[5] Trusted computing group. https://www.trustedcomputinggroup.org/.

[6] Youtube hijacking: A RIPE NCC RIS case study. http://www.ripe.net/news/study-youtube-hijacking.html.

[7] N. Aaraj, A. Raghunathan, and N. K. Jha. Analysis and design of a hardware/software trusted platform module for embedded systems. *ACM Transactions on Embedded Computing Systems*, 8(1), 2008.

[8] W. Aiello, J. Ioannidis, and P. McDaniel. Origin authentication in interdomain routing. In *Proc. of the ACM CCS*, pages 165–178, 2003.

[9] A. Beimel and B. Chor. Universally ideal secret sharing schemes. *IEEE Trans. on Info. Theory*, 40(3), 1994.

[10] J. Caballero, T. Kampouris, D. Song, and J. Wang. Would diversity really increase the robustness of the routing infrastructure against software defects? In *Proc. of the ISOC NDSS*, 2008.

[11] N. Feamster and H. Balakrishnan. Detecting bgp configuration faults with static analysis. In *Proc. of the NSDI*, 2005.

[12] S. Goldberg, S. Halevi, A. D. Jaggard, V. Ramachandran, and R. N. Wright. Rationality and traffic attraction: Incentives for honest path announcements in BGP. In *Proc. of the ACM SIGCOMM*, pages 267–278, 2008.

[13] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working around BGP: An incremental approach to improving security and accuracy of interdomain routing. In *Proc. of the ISOC NDSS*, pages 75–85, 2003.

[14] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10(2):232–243, 2002.

[15] R. Gummadi, H. Balakrishnan, P. Maniatis, and S. Ratnasamy. Not-a-bot: Improving service availability in the face of botnet attacks. In *Proc. of the NSDI*, 2009.

[16] A. Haeberlen, I. Avramopoulos, J. Rexford, and P. Druschel. Netreview: Detecting bgp configuration faults with static analysis. In *Proc. of the NSDI*, 2009.

[17] X. Hu and Z. M. Mao. Accurate real-time identification of IP prefix hijacking. In *Proc. of the IEEE Symposium on Security and Privacy*, 2007.

[18] Y. Hu, A. Perrig, and M. Sirbu. SPV: Secure path vector routing for securing bgp. In *Proc. of the ACM SIGCOMM*, pages 179–192, 2004.

[19] J. Karlin, S. Forrest, and J. Rexford. Pretty good bgp: Improving BGP by cautiously adopting routes. In *Proc. of the IEEE ICNP*, pages 290–299, 2006.

[20] E. Keller, M. Yu, M. Caesar, and J. Rexford. Virtually eliminating router bugs. In *Proc. of the ACM CoNext*, 2009.

[21] S. Kent, C. Lynn, and K. Seo. Secure border gateway protocol. *IEEE JSAC*, 18(4):582–592, 2000.

[22] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. PHAS: a prefix hijack alert system. In *Proc. of the USENIX Security Symposium*, 2006.

[23] Q. Li, M. Xu, J. Wu, X. Zhang, Patrick P.C. Lee, and K. Xu. Enhancing the trust of internet routing with lightweight route attestation. *Tsinghua CS Technical Report*, 2010.

[24] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (BGP-4). *RFC 4271*, 2006.

[25] P. Reynolds, O. Kennedy, E. G. Sirer, and F. B. Schneider. Securing BGP using external security monitors. *Cornell University, Computing and Information Science, Technical Report TR2006-2065*, 2006.

[26] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. of Crypto*, pages 47–53, 1984.

[27] E. Shi, A. Perrig, and L. van Doorn. BIND: A fine-grained attestation service for secure distributed systems. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 154–168, 2005.

[28] L. Subramanian, V. Roth, I. Stoica, S.Shenker, and R.H. Katz. Listen and whisper: Security mechanisms for BGP. In *Proc. of the NSDI*, 2004.

[29] W. Sun, Z. Mao, and K. Shin. Differentiated bgp update processing for improved routing convergence. In *Proc. of the ICNP*, 2006.

[30] P.C. van Oorschot, T. Wan, and E. Kranakis. On inter-domain routing security and pretty secure BGP (psBGP). *ACM TISSEC*, 10(3):1–41, 2007.

[31] R. White. Through secure origin BGP. *The Internet Protocol Journal*, 6(3):15–22, 2003.

[32] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush. ispy: Detecting IP prefix hijacking on my own. In *Proc. of the ACM SIGCOMM*, 2008.

[33] M. Zhao, S.W. Smith, and D.M. Nicol. The performance impact of BGP security. *IEEE Network*, 19(6):42–48, 2005.