Using Resource Public Key Infrastructure for Secure Border Gateway Protocol

George Chang, Majid Arianezhad, and Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia, Canada
{gkchang@sfu.ca, arianezhad@live.com, ljilja@cs.sfu.ca}

***Abstract*—Border Gateway Protocol (BGP) is a widely used Internet routing protocol. While several security features have been introduced and implemented to prevent attacks and address routing instabilities, BGP remains vulnerable due to lack of integrity and authentication of BGP messages. BGP operations strongly depend on its security and attacks on BGP adversely influence packet routing. Given the importance of BGP security, several approaches have been developed to enhance security of BGP sessions. The Resource Public Key Infrastructure (RPKI), a specialized Public Key Infrastructure (PKI), was developed to help secure the Internet routing. It uses cryptographically verifiable statements to ensure that Autonomous Systems (ASes), the Internet resource holders, are certifiably linked to the routing information they generate thus resulting in a reliable routing origin. In this paper, we describe simulation results and a testbed developed for validating route origin.**

Keywords—Routing protocols; BGP; BGP security; resource public key infrastructure

I. BGP Security

The Border Gateway Protocol (BGP) [1] was developed and designed before the Internet environment became subjected to attacks, exploits, and routing vulnerabilities. The originally designed BGP lacks security countermeasures required to prevent intentional or accidental network errors. These attacks and errors cause ripple effects that propagate throughout the network resulting in potentially catastrophic routing disruptions. These attacks may include modifying, deleting, forging, or duplicating update messages, session hijacking, Distributed Denial of Service (DDoS) attacks, or IP spoofing [2]. Malicious attacks alone do not account for all security issues. Non-intentional and accidental errors also contribute to network instability. Misconfigured or faulty routers may also inject falsified information while being legitimate BGP speakers. BGP does not have native counter-measures against these attacks.

BGP has undergone many revisions and security improvements. However, it still fails in preventing global routing disruptions and traffic hijacking. Resource Public Key Infrastructure (RPKI), built based on the Public Key Infrastructure (PKI), formally validates the original Autonomous System (AS) of route announcements. The use of resource certificate ensures that only the owner (resource holder) has the authority to advertise its routes thus preventing hijacking of the route origin.

II. Methods for Securing BGP

As the Internet de facto inter-domain routing protocol, BGP is targeted and subjected to attacks. Over the years, security measures have been developed and methods implemented in attempts to make the Internet more secure. Several approaches for enhancing BGP security have been introduced:

A. Explicitly Configuring BGP Peers

This method explicitly sets identical configurations between neighboring peer routers so that one may specify connectivity to be limited between peers with the same configuration. The requirement for identical configurations of all BGP speakers prevents session establishments by routers with non-matching configurations. The BGP speakers use TCP port 179 for communication. Since BGP relies on TCP, the inherent TCP risks and vulnerabilities also affect BGP. To prevent attackers from spoofing BGP packets sent over TCP and from hijacking the session, strong sequence number randomization may be employed. Since the attacker would then need to guess the sequence number of the packet for message injection or modification, using strong sequence number randomization would make predicting or guessing the correct sequence or acknowledgment (ACK) number improbable [3].

B. Using BGP Session Shared Secrets

Using BGP session sharing for securing BGP involves using a shared key between both ends of the connection. This key is used to compare all incoming BGP packets that contain a 16-byte digest value created by the Message Digest 5 (MD5) algorithm. The recipient of the BGP packets will use its shared key to compute the digest and compare it with the digest contained in the packet. If a mismatch occurs, the recipient should not respond to the sender and will discard the packet [4], [5]. The MD5 algorithm provides authentication to packets and helps prevent spoofing. A shared secret key should be changed periodically and should be unique between peering sessions. While MD5 ensures integrity and prevents message duplication, it does not ensure the packet confidentiality. The management of the security keys for various sessions has also been proposed [6]. It has been observed that MD5 is "cryptographically broken and unsuitable for further use" [7].

C. Leveraging an IPsec Tunnel

Internet Protocol Security (IPsec) provides strong protection for message integrity and assist in prevention of Denial of Service (DoS). IPsec secures confidentiality of

messages and authenticity of peer sessions. It excels in peer-to-peer connectivity and communications. IPsec employs security keys and security protocols such as Authentication Header (AH), Encapsulating Security Payloads (ESP), and Internet Key Exchange (IKE) for key management [7], [9]. While IPsec is superior to MD5 in terms of its refreshed keys, it lacks in terms of resource utilization. IPsec consumes CPU resources and introduces overhead to the router due to its requirement that a link be established beforehand between peers. However, an attacker may still send large amount of spoofing packets with false authentication to a router. This results in increased resource utilization of a router's CPU and may slow or halt legitimate packets from being processed. If the router crashes, the attacker accomplishes its goal. The negative impact of utilizing IPsec tunnels in terms of resource consumption, higher maintenance, and troubleshooting difficulties made IPsec tunneling more suitable for point-to-point connections such as Virtual Private Networks (VPN). IPsec is not adequate to prevent widespread attacks because it is a session-based security method [10].

D. Using Loopback Addresses for BGP Peers

Loopback is a virtual interface that enables connectivity of a physical router to pass its inbound and outbound messages through virtual connections. The loopback prevents the attacker from gaining the physical source address. It also keeps the TCP session alive between two routers using loopback addresses when the physical connection is down. This security method prevents the disruption and/or hijacking of TCP sessions.

E. Controlling the Time-To-Live (TTL) of BGP Packets

This method, also known as the Generalized TTL-based Security Mechanism (GTSM) [11], prevents CPU overload based attacks as well as other resource utilization-based attacks. GTSM was devised by the IETF and relies a simple TTL mechanism that TTL spoofing is considered impossible or highly unlikely. The nature of BGP peering is, in most cases, direct or adjacent, which makes GTSM feasible. Most BGP pairings are direct connections between peers and only one hop away. The TTL field in an IP packet decreases by one every time the packed completes a hop. GTSM restricts the TTL field to a value above a threshold and drops the packets that have been through multiple hops and, thus, have a TTL value lower than the threshold. By setting the TTL field to 255 (the maximum number in the IPv4 header), the peers will receive and decrease the TTL value to 254. Therefore, by only accepting packets that contain an inbound TTL number of 255 and above, GTSM may prevent attacks that transmit massive non-neighbor originating packets by flooding. GTSM may protect BGP from remote attackers that send spoofed messages. However, it will not protect BGP against multi-hop or other type of attacks.

F. Filtering on the Peering Interface

As a best practice, filters placed on the router's interface should be configured to prevent unwanted packets from unknown origins. Filters should allow only neighboring peers to speak. For example, filters should allow TCP port 179 packets that are sourced from a neighboring peer via direct connection.

G. Using Link-Local Peering

Link-local peering employs the link local address of the neighboring routers as the IPv6 default address instead of the global address. This may prevent attackers from establishing a connection with the router. It is considered infeasible that attackers will obtain these addresses thus securing session authentication. When using Link-Local addressing, the next-hop address is set as the global IPv6 address because the Link-Local addresses are local and used exclusively within the Link-Local address' subnet. Therefore, for the router to perform global routing, a route map is needed to specify a global address as the next-hop address. Otherwise, the route would be dropped if peers from other subnets could not reach the specified next-hop destination [12]. The routers may also advertise both Link-Local address and the global address in the reachability information attribute. This is typical for two peers residing within one subnet.

The drawback of Link-Local peering is that configurations on both sides of the peering should be identical. A change in the configuration on one side would result in routing instability and route flapping. Since the Link-Local addresses are derived from the Media Access Card (MAC) address, hardware changes, such as changes in Network Interface Card (NIC), would require reconfiguration in both peers [13]. Any upgrades, maintenance modifications, or improvements would also need to be performed on both peers to prevent session loss or failure. The additional effort in implementing Link-Local peering may not yield significant improvements that could be achieved by other security methods.

H. Preventing Long AS Paths and Limiting the Number of Prefixes Received

An attacker may inflict damage to the network by prepending and sending unusually long AS paths in the update messages. These false announcements are then stored and computed by the receiving peers, wasting CPU and memory resources. Older models or ill-configured routers may be unable to handle these long AS paths and may start to flop and choke [14]. This results in the tear down of connections and sessions between peers. The damage continues as core routers that are capable of handling the long ASes will continue to propagate these AS paths to their peers, spreading the damage. The flood of these updates will cause global routing instability [15]. For example, in 2009, Supronet, a small Czech provider announced an AS path that is 251 times longer than usual to its backup provider. Many older routers responded to the announcement by justifying it as malformed and, thus, tore down their sessions with the speaker who propagated the message. The result was an excessive rate of 107,780 updates per-second worldwide within 8 minutes of the initial announcement by Supronet [16]. This illustrates that a single event may disrupt the global networking. The same effect may also be achieved using prefixes. Attacker may send large number of prefixes to cripple a router. The remedy is to limit either the number of prefixes or the length of AS paths that are accepted and, thus, prevent the attack of overloading a router's available resources. Many routers may also be configured to re-establish the lost session after a certain time interval to prevent the link from being lost forever.

I. Securing IGP

Since BGP relies on the TCP layer, it is vital to secure the transport layer. BGP relies on IGP to reach the next hop or peer and, hence, the security of IGP is essential . Performing vital countermeasures and implementing security technologies or algorithms such as IPsec and MD5 on IGP may prevent attackers from infiltrating and damaging the network.

J. Extreme Measures for Securing Communications between BGP Peers

Manually setting information during configuration may greatly reduce the security risks in BGP peering. Configurations such as disabling the Neighbor Discovery Protocol (NDP) for IPv6 may prevent attackers from launching a DoS attack [17]. Manually configuring the static IPv6 addresses on the interface is much safer. It is similar to establishing a link-local peering where one explicitly specifies the connections thus removing the NDP from configuration. This method may have hidden side effects and may result in additional costs and troubleshooting if routers are ill-configured.

III. BGP Routing Outages

Current BGP security mechanisms do not focus on route origin validation. BGP has no built-in methods to validate the origin of prefixes for route advertisements across the Internet. This implies that routers are prone to rogue routing information and will likely forward it to other routers. Consequently, if a route is hijacked or wrongly advertised, BGP would be unable to detect it and promptly react to avoid service interruption.

Over the years, there have been many incidents in routing outages [18]. Most of incidents involve trusting one or more transit carriers that will unknowingly advertise the incorrect or unauthorized routing information across the Internet. One example occurred on February 24, 2008 when the Pakistan Telecom (AS 17557) advertised a /24 route for YouTube (AS 36561) and hijacked the traffic. Since the /24 is a more specific route than /22 that YouTube advertises, part of the Internet chose to route the traffic to Pakistan Telecom, instead. This unauthorized global announcement of YouTube address space was the consequence of BGP's transitive trust model. The Pakistan Telecom's (17557) main transit provider PCCW (AS 3491) simply trusted Pakistan Telecom's routing information and re-advertised it to its peers without validation. Since there is also mutual trust between PCCW (3491) and its peers through transitive trust model, PCCW's peers did not verify the route origin [19]. The result was an outage of YouTube services for more than two hours. Although this incident may have been unintentional, it illustrates that without adequate route origin validation, attackers could achieve BGP routing outages. Origin validation for BGP is a mechanism that may resolve security issues. It employs uses RPKI to ensure that route advertisements are originated from the expected AS.

IV. Resource Public Key Infrastructure

The most common routing errors occur when an unauthorized holder of the address space announces a particular IP prefix. RPKI offers BGP origin validation to avoid errors in routing decisions. RPKI, also known as resource certification, is introduced by the Internet Engineering Task Force (IETF) and the Secure Inter-Domain Routing (SIDR) workings group to prevent route hijacking. RPKI is built on top of the well-established PKI system that uses a public key cryptographic technique to verify identities based on digital certificates. The certificate fields and the X.509 standard for RPKI are described in RFC 5280 [20]. In 2011, Réseaux IP Européens (RIPE) launched the RPKI system that enables Local Internet Registries (LIRs) to request a digital certificate for the Internet resources that they hold. Other Regional Internet Registries (RIRs) then established certificate systems to validate all resources they allocate or assign [21]. The RPKI system may be used to certify anything related to the Internet resource.

In order to participate in route origin validation system, network operators need to create their own Route Origin Authorizations (ROAs) that will validate route announcements. ROAs are cryptographically objects verifying ASes that are authorized to originate certain address space in BGP announcements across the Internet. An ROA contains three elements: the authorized AS number, the address space that an AS may originate, and the maximum length of the authorized address space. In RPKI system, route announcements maybe valid, invalid, or unknown [22]. Based on these states that indicate the status of a route, the network operator may perform routing decisions.

A. RPKI Validators

Participants in RPKI may use a validator tool that retrieves information from RIRs' repositories to improve BGP routing decisions [23], [24]. The recommended tools are from: RIPE Network Coordination Centre (RPKI Validator), Dragon Research Labs (rcynic Validator), and Raytheon BBN (RPSTIR Project). RIPE RPKI validator provides a web user interface for viewing, configuring, and querying both validated ROAs and previewing RPKI validity state of BGP announcements across the Internet. RPKI-capable routers may connect to the RIPE RPKI validator and transfer validated ROA datasets. RIPE RPKI validator is preconfigured to automatically download and validate with trust anchors from four RIRs: AFRINIC, APNIC, LACNIC, and RIPE. Obtaining the trust anchor for ARIN requires accepting ARIN's Relying Party Agreement and then manually adding arin.tal file to the RPKI validator.

The ROAs are designed to be published rather than be confidential. They are stored in repositories made available to all RIRs and Internet Service Providers (ISPs). There is no authentication for ROAs since the PKI only offers authorization verification [25].

RIPE NCC provides a Java application toolset that acts as a local cache validator. This application runs as a service and only requires a UNIX system to operate. It includes a variety of components and options available to network operators to monitor and validate BGP routes. Application support and training is available on the RIPE NCC webpage. This application was ARIN's first RPKI Validation tool [26].

RPKI is an elegant way of ensuring resource authentication, allowing peers to better understand route

announcements and make routing decisions. The easy implementation and maintenance makes RPKI a suitable measure to secure BGP. Software and online portals provided by RIRs such as ARIN and RIPE make managing RPKI easy for valid subscribers.

V. SIMULATIONS

We designed a simulation scenario as a proof of concept to investigate RPKI's scalability. The goal was to implement the RPKI system with the validator tool using actual RPKI cache server data in a virtual environment.

The simulation was conducted using Graphic Network Simulator 3 (GNS3) [27] with the virtual machine (VM) running Linux as the "RPKI Validator". The VM is hosted on Oracle VirtualBox running Ubuntu 14.04.2 Trusty 64bit version image. The RPKI validator tool was installed on the Ubuntu VM running the latest June 5, 2015 RIPE RPKI Validator tool v. 2.2. Virtual routers were two Cisco c7200 routers running IOS image version 15.2 that supports RPKI. The router images were imported into GNS3 running v. 1.3.7.

A. *Simulation Setup: Network Topology and Configurations*

The simulation's network topology is shown in Figure 1.

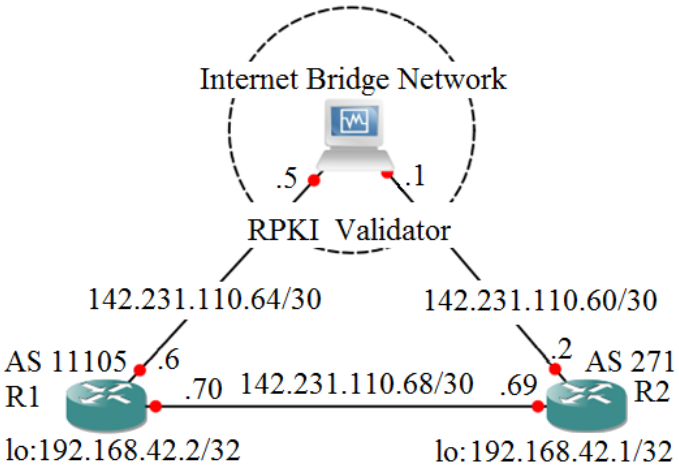

Figure 1: Simulation topology.

Two virtual routers labeled R1 and R2 are connected to each other and to the RPKI Validator through Ethernet connections. R1 was assigned an actual AS number 11105 that belongs to SFU while R2 was assigned BCNET's AS 271. By using actual AS numbers, actual data from the validator may be used to validate the advertised routes based on the AS numbers. The RPKI Validator requires the Internet connection to download the ROA resources to the VM. Simulation IP assignments are shown in Table I.

Router configurations were completed by assigning the rpki-loc-pref for each individual state. Both virtual routers were configured to connect to port 8282 of the RPKI Validator to download the ROA resources. R1 was setup to advertise three distinct routes with known states: valid, invalid, and unknown to R2. Router R2 would then validate the route advertised to it with live ROA data from the RPKI Validator.

TABLE I. SIMULATION IP ASSIGNEMNTS

Device	Interface	Prefix	IP Assignment	Description
R1 AS11105 SFU	ge-1/0	142.231.110.64/30	142.231.110.66	R1 router to Validator
	ge-1/1	142.231.110.68/30	142.231.110.70	R1 to R2
R2 AS271 BCNET	ge-1/1	142.231.110.60/30	142.231.110.62	R2 router to Validator
	ge-1/2	142.231.110.68/30	142.231.110.69	R1 to R2
RPKI Validator	eth0	Bridged network	Dynamic	Validator to the Internet
	eth1	142.231.110.64/30	142.231.110.65	Validator to R1
	eth2	142.231.110.60/30	142.231.110.61	Validator to R2

B. *Simulation Results*

The simulation results were conclusive. Router R2, that was receiving route advertisements from R1, was able to identify the validity of all routes received from R1 and to assign localpref numbers that were previously configured on router R2.

Valid State: advertised route 206.12.7.0 to R2 (AS 271) using an actual route validated for AS 11105. The state has been identified as "valid" with a localpref of 110 as configured:

```
R2#show ip bgp 206.12.7.0
BGP routing table entry for 206.12.7.0/24, version 3
Path: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 2
  11105
    142.231.110.70 from 142.231.110.70 (142.231.110.70)
      Origin IGP, metric 0, localpref 110, valid …
      Path 68DB44CC RPKI State valid
      Rx pathid: 0, tx pathid: 0x0
```

Invalid State: advertised an invalid route to R1 (AS 11105) from R2 (AS 271) using a randomly selected IP that was not in the RPKI database for both ASes. The state has been identified as "invalid" with a localpref of 90 as configured:

```
R1#sh ip bgp 193.175.146.0
BGP routing table entry for 193.175.146.0/24, version 0
Path: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 9
  271
    142.231.110.69 from 142.231.110.69 (142.231.110.69)
      Origin IGP, metric 0, localpref 90, valid …
      Path 682CAF34 RPKI State invalid
      Rx pathid: 0, tx pathid: 0
```

Invalid State: advertised an unknown route to R2 (AS 271) from R1 (AS 11105) using a randomly selected IP that was not in the RPKI database for both AS. The state has been identified as "not found" with a localpref of 100 as configured:

```
R2#sh ip bgp 6.0.0.0
BGP routing table entry for 6.0.0.0/8, version 5
Path: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 10
  11105
    142.231.110.70 from 142.231.110.70 (142.231.110.70)
      Origin IGP, metric 0, localpref 100, valid …
      Path 68DB4424 RPKI State not found
      Rx pathid: 0, tx pathid: 0x0
```

The simulations illustrate that RPKI may be easily implemented in deployed networks and that the simulations were capable of downloading actual ROA resources in a virtual

environment. The simulations may be useful to investigate further BGP vulnerabilities and to secure the Internet.

VI. DEPOLYED TESTBED

We have also built a testbed with the objectives to observe the states of route announcements and verify the effects of routing policies in RPKI BGP. The setup closely resembles the simulation's topology, as shown in Figure 2. Both routers are connected to a local cache validator that downloads the ROA dataset. The SFU and BCNET BGP speakers announce to each other globally routable prefixes 192.67.9.0/24 and 206.12.7.0/24 to each other. The testbed includes two routers and one local cache validator. Two logical routers were instantiated using JunOs software installed on the SFU and BCNET test equipment. JunOs software partitions a single router into multiple logical devices performing independent routing tasks. The IP assignments for the routers and the local cache validator are shown in Table II.

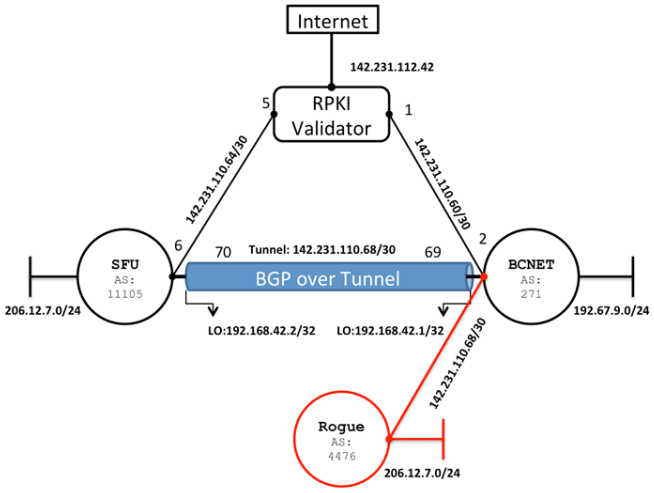

Figure 2: Logical topology of the deployed testbed.

TABLE II. IP ASSIGNEMNTS FOR THE SIMUALATION

Device	Interface	Prefix	IP Assignment	Description
SFU test router AS 11105	ge-0/0/0	142.231.110.64/30	142.231.110.66	SFU router to Validator
	lo0	192.168.42.2/32	192.168.42.2	Loopback
		206.12.7.0/24	206.12.7.1	SFU prefix for ROA
	lt-0/2/10	142.231.110.68/30	142.231.110.70	Tunnel
R2 AS271 BCNET	ge-0/0/0	142.231.110.64/30	142.231.110.66	BCNET router to Validator
	lo0	192.168.42.1/32	192.168.42.1	Loopback
		192.67.9.0/24	192.67.9.1	BCNET prefix for ROA
	ge-1/2	142.231.110.68/30	142.231.110.69	Tunnel
RPKI Validator	eth0	142.231.112.0/24	142.231.112.42	Validator to outside interface
	eth1	142.231.110.60/30	142.231.110.61	Validator to BCNET
	eth2	142.231.110.64/30	142.231.110.65	Validator to SFU

A. Testbed Architecture

An Ubuntu VM acts as a local cache validator. SFU and BCNET have obtained IP resources from ARIN and have distinct online accounts with ARIN's website. Each account in ARIN is linked to an administrator. Account holders in ARIN can manage and certify resources, such as IPv4 and IPv6 addresses. SFU and BCNET have created resource certificates and generated distinct ROA key pairs for their IP prefixes.

By default, the RIPE NCC RPKI validator application includes the Trust Anchor Locator (TAL) files for four RIRs: AFRINIC, APNIC, LACNIC, and RIPE NCC. Upon accepting ARIN's Relying Party Agreement, ARIN's TAL (ARIN's public key) is sent to the recipient's email address. A network operator needs to create and save ARIN's TAL. The validator service automatically loads files matching the *.tal on startup.

The RPKI validator machine has three interfaces: eth0, eth1, and eth2. The interface eth0 (outside interface) connects the validator to the Internet. Interfaces eth1 and eth2 are connected to the BCNET's and the SFU's routers, respectively. An important security practice is to configure firewall of the validator machine to accept connections from anticipated combinations of IP addresses and port numbers. In the deployed testbed, the validator only accepts TCP connections from 142.231.110.62:8282 and 142.231.110.66:8282 to eth1 and eth2, respectively.

There are several connection mechanisms between virtual routers: logical tunnel interfaces, rib-group, instance-import, and next-table. Logical tunnels are point-to-point interfaces that carry traffic between virtual routers. In the SFU-BCNET testbed, logical tunnel interfaces (lt-0/2/10) were used to form BGP peering between two virtual routers. We assigned the actual SFU AS number (AS 11105) and BCNET AS number (AS 271) to the logical routers, as in the simulations.

Each router has a loopback interface that may be assigned multiple IP addresses. Loopbacks were assigned invalid IP addresses (192.168.42.1 and 192.168.42.2) that were required to create the logical tunnel interfaces (lt-0/2/10). Moreover, in order to announce prefixes to routers, 206.12.7.1 (SFU prefix 206.12.7.0/24 with ROA) and 192.67.9.1 (BCNET prefix 192.67.9.0/24 with ROA) were assigned to SFU and BCNET's loopback connection, respectively.

B. Verifying Origin Validation

RPKI commands *show validation session*, *show validation statistics*, and *show validation database* may be performed on the router to verify the routes. RPKI command output:

```
tr1.vncv1> show validation session detail
Session 142.231.110.61, State: up, Session index: 2
  Group: BCNET_VALIDATOR, Preference: 200
  Local IPv4 address: 142.231.110.62, Port: 8282
  Refresh time: 300s
  Hold time: 900s
  Record Life time: 900s
  Serial (Full Update): 441
  Serial (Incremental Update): 441
    Session flaps: 2
    Session uptime: 1w0d 10:11:12
    Last PDU received: 00:01:29
    IPv4 prefix count: 7078
    IPv6 prefix count: 1106

tr1.vncv1> show validation statistics
Total RV records: 8190
Total Replication RV records: 8190
  Prefix entries: 7815
  Origin-AS entries: 8190
Memory utilization: 1590149 bytes
Policy origin-validation requests: 6
  Valid: 2 Invalid: 2   Unknown: 2
```

```
BGP import policy reevaluation notifications: 3
  inet.0, 3  inet6.0, 0
tr1.vncv1> show validation database
RV database for instance master
Prefix          Origin-AS Session       State   Mismatch
2.0.0.0/12-16   3215 142.231.110.61     valid
2.0.0.0/16-16   3215 142.231.110.61     valid
2.1.0.0/16-16   3215 142.231.110.61     valid
2.2.0.0/16-16   3215 142.231.110.61     valid
```

Verification of the applied policy was performed by setting local preferences for valid, invalid, and unknown states to 110, 90, and 100, respectively. As shown in Figure 2, a rogue test logical router was installed to verify the policy of an "invalid" state. The rogue router announced SFU's prefix, which originated from an invalid AS (AS 4476), to the BCNET router. The output of *show route protocol bgp validation-state invalid* and *show route 206.12.7.0* statements indicate that BCNET router recognized the invalid AS 4476 as expected:

```
tr1.vncv1> show route protocol bgp validation-state valid
inet.0: 13 destinations, 14 routes (13 active, 0 holddown,
0 hidden)
+ = Active Route, - = Last Active, * = Both

206.12.7.0/24*[BGP/170] 3w6d 05:23:33, localpref 110
              AS path: 11105 I, validation-state: valid
              > to 142.231.110.70 via lt-0/2/10.69

tr1.vncv1> show route protocol bgp validation-state invalid
inet.0: 13 destinations, 14 routes (13 active, 0 holddown,
0 hidden)
+ = Active Route, - = Last Active, * = Both
206.12.7.0/24 [BGP/170] 3d 08:00:09, localpref 90
              AS path: 4476 I, validation-state: invalid
              > to 142.231.110.66 via lt-0/3/10.65

tr1.vncv1> show route 206.12.7.0
inet.0: 13 destinations, 14 routes (13 active, 0 holddown,
0 hidden)
+ = Active Route, - = Last Active, * = Both
206.12.7.0/24*[BGP/170] 3w6d 05:27:15, localpref 110
              AS path: 11105 I, validation-state: valid
              > to 142.231.110.70 via lt-0/2/10.69
              [BGP/170] 3d 08:03:15, localpref 90
              AS path: 4476 I, validation-state: invalid
              > to 142.231.110.66 via lt-0/3/10.65
```

RPKI BGP does not preserve the authenticity and integrity of the AS path attribute carried in a BGP message. BGP speakers may insert bogus routing information and, thus, may cause widespread disruption. An important future improvement would be to enable BGP speakers to verify that the ordering sequence of ASes in the path attribute represent the sequence of ASes in the network layer reachability information (NLRI).

VII. Conclusion

Managing RPKI is easy with the existence of online portals, software, and tools provided by the RIRs such as ARIN and RIPE. Juniper JunOs v. 12.2 and Cisco fully support RPKI. In order to examine the functionality of RPKI BGP, we implemented a testbed using the JunOs software and RIPE RPKI validator software products. The experimental results show that RPKI BGP may provide protection against route origin hijacks.

Acknowledgment

The authors would like to acknowledge the contribution of M. Hay, D. McWilliam, and M. Gregory from BCNET for valuable assistance in setting up the testbed.

References

[1] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," *IETF RFC 1771*, Mar. 1995.

[2] S. Murphy, "BGP Security Vulnerabilities Analysis," *IETF RFC 4272*, Jan. 2006.

[3] Progress Toward Security the Routing Infrastructure [Online]. Available: http://www.cyber.st.dhs.gov/public/CATCH/Murphy.pdf.

[4] CERT Advisory CA-2001-09, "Statistical Weaknesses in TCP/IP Initial Sequence Numbers" [Online]. Available: http://www.cert.org/advisories/CA-2001-09.html.

[5] A. Heffernan, "Protection of BGP Sessions via the TCP MD5 Signature Option," *IETF RFC 2385*, Aug. 1998.

[6] S. Turner and L. Chen, "Updated Security Consideration for the MD5 Message-Digest and the HMAC-MD5 Algorithms," *IETF RFC 6151*, Mar. 2011.

[7] M. Leech, "Key Management Consideration for the TCP MD5 Signature Option," *IETF RFC 3562*, July 2003.

[8] C. Kaufman, "Internet Key Exchange (IKEv2) Protocol," *IETF RFC 4306*, Dec. 2005.

[9] S. Kent, "IP Authentication Header," *IETF RFC 4302*, Dec. 2005.

[10] S. Kent, "IP Encapsulating Security Payload (ESP)," *IETF RFC 4303*, Dec. 2005.

[11] K. Butler, P. McDaniel, T. R. Farley, and J. Rexford, "A survey of BGP security issues and solutions," *IEEE Journal on Selected Areas in Communications*, vol. 98, no. 1, pp. 5–10, Jan. 2010.

[12] V. Gill, J. Heasley, D. Meyer, P. Savola, and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)," *IETF RFC 5082*, Oct. 2007.

[13] P. Marques, and F. Dupont, "Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing," *IETF RFC 2545*, Mar. 1999.

[14] IPv6 Configuration [Online]. Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-2/ipv6_autoconfig.html

[15] DDoS and Security Reports: The Arbor Networks Security Blog [Online]. Available: http://ddos.arbornetworks.com/2009/02/ahh-the-ease-of-introducing-global-routing-instability/.

[16] European Network and Information Security Agency: Good Practices in Resilient Internet Interconnection [Online]. Available: http://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-infrastructure-and-services/inter-x/resilience-of-interconnections/report/.

[17] Reckless Driving on the Internet [Online]. Available: http://www.renesys.com/2009/02/the-flap-heard-around-the-world/.

[18] I. Gashinsky, J. Jaeggli, and W. Kumari, "Operational Neighbor Discovery Problems," *IETF RFC 6583*, Mar. 2012.

[19] Pakistan hijacks [Online]. Available: YouTube http://www.renesys.com/2008/02/pakistan-hijacks-youtube-1/.

[20] D. Cooper et al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," *IETF RFC 5280*, May 2008.

[21] M. Lepinski and S. Kent, "An Infrastructure to Support Secure Internet Routing," IETF RFC 6480, Feb. 2012.

[22] G. Huston and G. Michaelson, "Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)," IETF RFC 6482, Feb 2012.

[23] R. Bush and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol," IETF RFC 6810, Jan. 2013.

[24] P. Mohapatra, J. Scudder, D. Ward, R. Bush, and R. Austein, "BGP Prefix Origin Validation," IETF RFC 6811, Jan. 2013.

[25] Resource Public Key Infrastructure (RPKI) [Online]. Available: https://www.arin.net/resources/rpki/index.html.

[26] M. Lepinski, S. Kent, and D. Kong, "A Profile for Route Origin Authorizations (ROAs)," *IETF RFC 6482*, Feb. 2012.

[27] Graphic Network Simulator 3 (GNS3) [Online]. Available: https://www.gns3.net.