

**TOWSON UNIVERSITY  
OFFICE OF GRADUATE STUDIES**

**Real-Time Verification Tool of BGP Routing Information for Preventing  
Inter-Domain Routing Misbehavior**

**by**

**Je-Kuk Yun**

**A Dissertation**

**Presented to the faculty of**

**Towson University**

**in partial fulfillment**

**of the requirements for the degree**

**Doctor of Science in Information Technology**

**Department of Computer and Information Sciences**

**Towson University  
Towson, Maryland 21252**

**May 2015**

**TOWSON UNIVERSITY  
OFFICE OF GRADUATE STUDIES**

**DISSERTATION APPROVAL PAGE**

This is to certify that the dissertation prepared by Je-Kuk Yun , entitled "Real-Time Verification Tool of BGP Routing Information for Preventing Inter-Domain Routing Misbehavior," has been approved by the thesis committee as satisfactorily completing the dissertation requirements for the degree of Doctor of Science in Information Technology.



Chairperson, Dissertation Committee, Dr. Yanggon Kim

4/17/2015

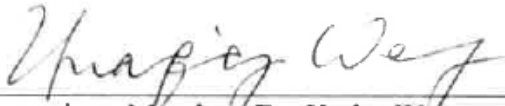
Date



Committee Member, Dr. Robert Hammell

4/17/2015

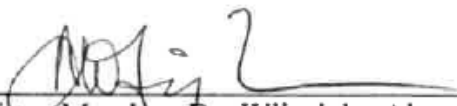
Date



Committee Member, Dr. Kathy Wang

4/17/2015

Date



Committee Member, Dr. Wijesinha Alexander

4/17/15

Date



Committee Member, Dr. Wei Yu

4/17/2015

Date



Dean of Graduate Studies

5-5-15

Date

© 2015 By Je-Kuk Yun  
All Rights Reserved

## ACKNOWLEDGEMENTS

This dissertation would not have been possible without the help of so many people in so many ways. I am most grateful to my dissertation advisor, Dr. Yanggon Kim, whose understanding, suggestions, generous guidance, enthusiasm and encouragement made it possible for me to continue to working on this topic farther than I thought I could go. Since the time I came to Towson as a master's student, I have been guided and encouraged by him to be an independent researcher as a doctorate student.

I would like to thank Dr. Ohoe Kim for helping me know what I love to do. Throughout every other week's creative workshop for 4 years, he provided me with the foundation for becoming a creative researcher, who can solve problems in different and creative ways.

I would also like to thank Dr. Hyeong-Ah Theresa Choi at George Washington University for giving me an opportunity to participate in many practical projects for about 4 years. Those projects brought me invaluable experience, cooperation with team members, and strong confidence to solve any problems.

My thanks go to the committee members, Dr. Robert Hammell, Dr. Kathy Wang, Dr. Wijesinha Alexander, and Dr. Wei Yu for their guidance, encouragement and help throughout this dissertation. Also, I would like to thank our lab members for many valuable comments that improved the contents of the dissertation and presentation. Of course, no acknowledgment would be complete without giving thanks to my parents. Both have given me a good foundation on which to build my life. They've taught me about self-respect and about how to be independent. Both have always expressed how proud they are of me and how much they love me. I too am proud of them and love them

very much. In addition, I would like to thank my sister and brother-in-law for supporting me in many ways.

Above all, I would like to thank God, the Almighty, for always being with me.

*I dedicate this dissertation to my family  
for their faithful support and everlasting love.*

*I deeply love you all.*

## **ABSTRACT**

### **Real-Time Verification Tool of BGP Routing Information for Preventing Inter-Domain Routing Misbehavior**

**Je-Kuk Yun**

The Border Gateway Protocol (BGP) is an Inter-domain routing protocol that has gradually evolved over the past few decades. The main functionality of BGP is to exchange Network Layer Reachability Information (NLRI) using a BGP update message between autonomous systems (ASes) where BGP routers find a better path to the destination using NLRI. However, BGP is highly vulnerable to hijacking attacks because BGP itself does not have a mechanism to validate the BGP message. Two well-known types of hijackings are IP prefix hijacking and AS path hijacking. As the number of IP hijacking incidents has increased, many IP hijacking monitoring tools have been implemented. However, none of the monitoring tools can directly control the data plane of BGP routers. Therefore, network administrators should protect their routers by using command line interface when the network administrator receives any warning from BGP hijacking monitoring tools. As the number of routers and prefixes continuously increases, checking the routing information in their routers manually is one of the big burdens on the administrators. In addition, when IP hijacking occurs, it is very important for the administrator to quickly block the bogus prefixes. Otherwise, thousands of traffic will be transferred to the wrong destination within a very short moment. We extended Quagga-SRx so that the Quagga-SRx can send a BGP update message including an opaque extend community to other iBGP

peers for notifying bogus IP prefixes after detecting abnormal IP prefixes. As a result of this, the other iBGP peers can recognize bogus IP prefixes by accepting the BGP update message that includes the opaque extend community, and the iBGP peers can automatically block the bogus prefixes if the iBGP peers have the ability to process the opaque extend community. Therefore, when IP hijacking occurs, the bogus prefixes can be blocked automatically and quickly, which makes the ASes more secure. Even though many solutions are proposed to prevent IP hijacking, such as RPKI, BGPmon, Argus, and PHAS, all of the solutions except RPKI proposed so far can protect IP hijacking only through the origin validation. However, the origin validation cannot prevent AS path hijacking. In order to protect AS path hijacking, the SIDR working group proposed the RPKI using BGPSEC, but BGPSEC is currently a work in progress. So, we propose Secure AS\_PATH BGP (SAPBGP) in which we monitor AS\_PATH in update messages whether each AS in AS\_PATH are connected to each other based on our policy database collected from RIPE NCC repository. Our analysis shows 4.57% of AS\_PATH is invalid and 95.43% of AS\_PATH is valid from the fifteenth of April in 2014 to the eighth of June in 2014. In addition, the performance test verifies that the SAPBGP can process all of the live BGP messages coming from BGPmon in real time. The invalid ASes from the experiment could be either the AS does not configure policies or the AS\_PATH was manipulated by hijackers. For the precise experiment of the policy based AS\_PATH validation, every router needs to configure policies against its peers.



## TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iv
ABSTRACT .....	vii
TABLE OF CONTENTS .....	ix
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xii
LIST OF ABBREVIATIONS .....	xiv
<b>Chapter 1. Introduction.....</b>	<b>1</b>
<b>1.1 Contributions .....</b>	<b>5</b>
<b>1.2 Dissertation outline .....</b>	<b>6</b>
<b>Chapter 2. Related Work .....</b>	<b>7</b>
<b>2.1 Design and Operation of BGP .....</b>	<b>8</b>
<b>2.2 BGP's Vulnerabilities .....</b>	<b>14</b>
<b>2.2.1 IP hijacking .....</b>	<b>14</b>
<b>2.2.2 AS path hijacking.....</b>	<b>16</b>
<b>2.2.3 Peer spoofing.....</b>	<b>18</b>
<b>2.3 BGP information validation.....</b>	<b>18</b>
<b>2.3.1 Origin validation.....</b>	<b>18</b>
<b>2.3.2 Path validation .....</b>	<b>23</b>
<b>2.4 BGP Security Architectures .....</b>	<b>30</b>
<b>2.4.1 Secure BGP(S-BGP) and Secure Origin BGP(SO-BGP) .....</b>	<b>30</b>
<b>2.4.2 Pretty Secure BGP (psBGP).....</b>	<b>31</b>
<b>2.4.3 Inter-domain Route Validation (IRV).....</b>	<b>32</b>
<b>2.4.4 Pretty Good BGP (pgBGP).....</b>	<b>33</b>
<b>2.4.5 Secure Path Vector (SPV) .....</b>	<b>33</b>
<b>2.5 Existing security tools.....</b>	<b>34</b>
<b>2.6 BGP Policy Database .....</b>	<b>36</b>
<b>Chapter 3. The BGPMAPS .....</b>	<b>36</b>
<b>3.1 Overview of the BGPMAPS .....</b>	<b>36</b>
<b>3.2 Architecture of the BGPMAPS.....</b>	<b>38</b>
<b>3.3 Operational requirements.....</b>	<b>39</b>
<b>3.4 Implementation.....</b>	<b>40</b>
<b>3.5 Experiments.....</b>	<b>43</b>
<b>3.6 Performance test.....</b>	<b>46</b>

<b>Chapter 4. The SAPBGP</b> .....	51
<b>4.1 Overview of the SAPBGP</b> .....	51
<b>4.2 Monitor AS Path connections</b> .....	52
<b>4.3 Architecture of the SAPBGP</b> .....	53
<b>4.4 Experiments</b> .....	61
<b>4.5 Performance test</b> .....	63
<b>4.6 Result analysis</b> .....	65
<b>Chapter 5. Conclusion and Future Research</b> .....	72
<b>5.1 Conclusion</b> .....	72
<b>5.2 Future Research</b> .....	74
<b>REFERENCES</b> .....	76
<b>CURRICULUM VITAE</b> .....	83

## LIST OF TABLES

Table 1. Path Attribute types .....	11
Table 2. AS 6059's ROA.....	22
Table 3. The number of prefixes in whole AS.....	47
Table 4. The number of neighbors and prefixes in whole AS .....	48
Table 5. 32 bits AS number allocation above 65535 .....	55
Table 6. 9 organizations who participated in the BGPmon project .....	59
Table 7. The number of BGP Update Messages from BGPmon Project .....	61
Table 8. AS_PATH Validation Time To Process One BGP Update Message .....	65
Table 9. The Comparison of the Results .....	66
Table 10. Analysis of the number of AS hops in AS_PATH attributes .....	70
Table 11. Top 20 1-hop hijacking candidates.....	71

## LIST OF FIGURES

Figure 1. Path Attributes .....	9
Figure 2. Attribute Type (2 Bytes) .....	10
Figure 3. MED example.....	13
Figure 4. IP prefix hijacking .....	15
Figure 5. Manipulating AS_PATH attributes .....	17
Figure 6. Hierarchy of the RPKI.....	19
Figure 7. Certificate Chain.....	20
Figure 8. ROA Format .....	21
Figure 9. Scenario of IP hijacking .....	22
Figure 10. IPv4 covered by ROAs.....	23
Figure 11. 1-hop hijacking.....	24
Figure 12. BGPsec_Path Attribute.....	26
Figure 13. Example of the pCount attribute.....	27
Figure 14. Protecting the 1-hop hijacking by BGPsec.....	28
Figure 15. Inter-domain Route Validation .....	32
Figure 16. The architecture of the BGPMAPS .....	39
Figure 17. The update message format .....	41
Figure 18. A sequence diagram of the BGPMAPS .....	43
Figure 19. Topology for simulation .....	44
Figure 20. Ex-Quagga-SRx BGP table .....	45
Figure 21. Normal BGP table .....	46
Figure 22. R1-1 BGP table.....	46
Figure 23. Result of the performance test.....	48
Figure 24. The number of required command lines of reconfiguration.....	50
Figure 25. A scenario of manipulating a BGP message .....	52
Figure 26. The architecture of the SAPBGP.....	54
Figure 27. sample of the routing policies by the RIPE NCC API.....	56
Figure 28. A screen capture of the policy table.....	58
Figure 29. BGPmon sample data .....	60
Figure 30. Ratio of ASes that registered BGP routing policies .....	62
Figure 31. # of ASes that registered BGP policies.....	63
Figure 32. The result of the performance test for the AS_PATH validation .....	64
Figure 33. The result of the AS_PATH monitoring experiment that includes duplications.	

.....	67
Figure 34. A portion of the routing policy table of invalid ASes that include duplications	
.....	68
Figure 35. The result of the AS_PATH monitoring experiment that does not include duplications	
.....	69
Figure 36. A portion of the routing policy table of invalid ASes that does not include duplications	
.....	70
Figure 37 distributed database for BGP routing policy information	
.....	75

## LIST OF ABBREVIATIONS

AfNIC	African Network Information Centre
APNIC	Asia Pacific Network Information Centre
ARIN	American Registry for Internet Numbers
AS	Autonomous System
ASN	Autonomous System Number
BGP	Border Gateway Protocol
BGP-4	Border Gateway Protocol 4
BGPMAPS	Border Gateway Protocol Monitoring, Alarming, and Protecting System
CA	Certification Authority
CIDR	Classless Inter-Domain Routing
DHCP	Dynamic Host Configuration Protocol
DDoS	Distributed Denial of Service
DNS	Domain Name System
DOS	Denial of Service
eBGP	Exterior Border Gateway Protocol
EGP	Exterior Gateway Protocol
FIB	Forwarding Information Base
GTSM	Generalized TTL Security Mechanism
IANA	Internet Assigned Number Authority
iBGP	Internal Border Gateway Protocol
ICANN	Internet Corporation for Assigned Names and Numbers
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol (e.g., iBGP, OSPF, RIP)
IP	Internet Protocol
IPsec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IRV	Interdomain Route Validation
ISO	International Organization for Standardization
ISP	Internet Service Provider

IT	Information Technology
ITL	Information Technology Laboratory
LAN	Local Area Network
LANIC	Latin America and Caribbean Network Information Centre
MD5	Message-Digest 5
MED	Multi-Exit Discriminator
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
OSPF	Open Shortest Path First
pgBGP	Pretty Good Border Gateway Protocol
psBGP	Petty Secure Border Gateway Protocol
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure (X.509)
RA	Route Attestation
RFC	Request for Comment
RFD	Route Flap Damping
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPE NCC	Réseaux IP Européens Coordination Centre (RIPE, French for "European IP Networks")
RIS	Routing Information Service
RPKI	Resource Public Key Infrastructure
SAPBGP	Secure AS_PATH BGP
S-BGP	Secure Border Gateway Protocol
soBGP	Secure Origin Border Gateway Protocol
SPV	Secure Path Vector
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TTL	Time to Live

# Chapter 1. Introduction

Internet is a collection of networks, called Autonomous Systems (ASes), in which routers are connected to each other and personal devices that need Internet connections are connected to the router. Each AS is managed and owned by legal Internet Service Providers (ISPs) or companies, such as Verizon, Sprint, Comcast, Google, etc., and they belong to one of 5 Regional Internet Registries (RIRs): African Network Information Centre (AfriNIC), American Registry for Internet Numbers (ARIN), Asia-Pacific Network Information Centre (APNIC), Latin America and Caribbean Network (LACNIC), and Reseaux IP Europeens Network Coordination Centre (RIPE NCC). In order to provide the allocation of IP addresses and AS numbers, the Internet Registry (IR) hierarchy was designed, and the root of the hierarchy is the Internet Assigned Numbers Authority (IANA) [1]. The IANA delegates RIRs to allocate IP addresses and AS numbers to ISPs or end-user organizations. When ASes want to announce their IP addresses to other ASes, Border Gateway Protocol (BGP) is used.

The first version of BGP was released in the late 1980s [2]. In 1990, a second version of BGP [3] was released to remove the topological constraints and a third version of BGP [4] was released to optimize the exchange of information regarding previously reachable routes in 1991. Finally, the fourth version of BGP [5] was introduced in 1994 and revised in 1995 [6] and 2006 [7]. Within one AS, there should be at least one BGP router to exchange Network Layer Reachability Information (NLRI) to other ASes and BGP



routers should create a BGP connection between them. Once two BGP routers are connected to each other, we called them BGP peers. If a BGP router communicates with a BGP router that is located in the same AS, we called the BGP router iBGP peer, and if a BGP router communicates with a BGP router that is located in a different AS, we called it an eBGP peer. BGP routers announce their blocks of IP addresses to neighbors, and then the neighbors update their routing table and forward the updated routing information to other neighbors. In that way, BGP peers keep updating their routing table whenever new blocks of IP addresses are added or existing blocks of IP addresses are removed. As a result, BGP router can transfer Internet packets to their destination based on routing table.

The initial design of BGP was a fully trust-based system. So, BGP itself does not have mechanisms to verify whether a route is valid or not because BGP routers completely trust other BGP routers. This lack of consideration of BGP vulnerabilities often causes severe failures of Internet service provision [8]. Once a hijacking BGP router announces bogus blocks of IP addresses to BGP peers, the BGP peers transfer Internet traffic to the hijacking BGP router if the destination IP address is matched and the number of hops is shorter than the others. We call this threat of failures IP hijacking.

Such a failure occurred on the twenty fifth of April in 1997 by a misconfigured router that advertised incorrect prefixes and announced AS 7007 as the origin of them. As a result, it created a routing black hole for almost two hours [9]. Similar events occurred on the twenty second of January

in 2006, when Con Edison (AS 27506) stole several important prefixes by misconfiguring them [10]. On Christmas Eve, 2004, TTNNet in Turkey (AS 9121) advertised the entire prefixes on the Internet so that every route came to them rather than to correct destinations [11].

The most well-known IP hijacking is the YouTube hijacking by Pakistan Telecom (AS17557) on the twenty fourth of February in 2008 [12]. In response to a government order to block YouTube access within their ASes, Pakistan Telecom announced a more specific prefix than YouTube's prefix. Then, one of Pakistan Telecom's upstream providers, PCCW Global (AS3491), forwarded the announcement to other neighbors. As a result of this, YouTube traffic from all over the world was misled to Pakistan Telecom (AS17557) for two hours. In addition, The Dell SecureWorks Counter Threat Unit (CTU) research team discovered a repeated traffic hijacking to Bitcoin mining sites between February and May 2014. Compromised networks belonged to Amazon, Digital Ocean, OVH, etc. The attacker hijacked cryptocurrency miners' traffic and earned an estimated \$83,000 [13]. Furthermore, AS 23274, owned by China Telecom, announced approximately 50,000 prefixes, which were registered to other ASes in 2010. The reason the incident was magnified is because China Telecom was the 11th largest Internet provider. If small ISPs hijack a large part of the Internet, they don't have the capacity to deal with a huge amount of traffic. China Telecom, however, has the capability to operate under such traffics, and redirect its desired destination. The incident was not recognized for 18 minutes [14]. In order to solve the IP hijacking, many

studies were conducted, such as RPKI [15], BGPmon [16], Argus [17], and PHAS [18].

While there are many studies on the IP hijacking, few studies have been researched about an AS path hijacking. There was some misdirected network traffic that was suspected of the man-in-the-middle (MITM) attack in 2013 observed by Renesys. In February 2013, global traffic was redirected to Belarusian ISP GlobalOneBel before its intended destination and it occurred on an almost daily basis. Major financial institutions, governments, and network service providers were affected by this traffic diversion in several countries including the U.S. From the thirty first of July to the nineteenth of August in 2013, Icelandic provider Opin Kerfi announced origination routes for 597 IP networks owned by a large VoIP provider in the U.S through Siminn which is one of the two ISPs that Opin Kerfi has. However, this announcement was never propagated through Fjarskipti which is the other one of the two ISPs. As a result, network traffic was sent to Siminn in London and redirected back to its intended destination. Several different countries in some Icelandic autonomous systems and belonging to the Siminn were affected. However, Opin Kerfi said that the problem was the result of a bug in the software and had been resolved [19]. A root cause of BGP hijacking can be discovered by empirical data analysis using BGP updates from Routeviews, RIB from iPlane project, paths from traceroute, etc. However, proving a malicious intent is hardly possible. According to this research, China Telecom incident is most likely caused by a routing table leak [19].

In order to protect the AS path hijacking, the AS\_PATH should not be manipulated. However, the BGP itself cannot check whether the AS\_PATH has been changed or not. If a routing hijacker manipulates the AS\_PATH in a BGP message that is sent by another router and forwards the manipulated BGP message to other neighbors, the neighbors who receive the manipulated BGP message can be a victim of AS path hijacking. Only Secure Inter-Domain Routing (SIDR) working group proposed the RPKI using BGPSEC to validate AS\_PATH, but BGPSEC is currently a work in progress [20], [21]. In addition, a study propounds that BGP armed with BGPSEC cannot be secured because of BGP's fundamental design [22], [23].

We propose Secure AS\_PATH BGP (SAPBGP) in which the SAPBGP constructs its own policy-based database by collecting RIPE NCC repository and checks the AS\_PATH attribute in BGP update messages whether the ASes listed in the AS\_PATH attributes are actually connected or not. For the validation test with the real BGP messages, the SAPBGP receives live BGP streams from BGPmon project [24], [25]. In addition, we conduct the performance test of the SAPBGP to measure the duration of the validation with the live BGP messages.

## **1.1 Contributions**

The main contributions of the dissertation are as follows:

1. A BGP monitoring, alarming, and preventing system has been developed for notifying network administrators of invalid IP prefixes by
  - extending Quagga-SRx (ex-Quagga-SRx) that can handle an opaque extended community attribute and store validation results for history purposes.
  - developing Alarm Server and Alarm System so that network administrators can be notified of invalid prefixes even though either BGP router cannot handle opaque extended community attribute or the network administrators manages BGP router in different ASes.
2. Performance analysis of securing iBGP peers by sending BGP update message including opaque extended community.
3. A novel approach to protect BGP routers from AS\_PATH hijacking by using policy-based database.
4. Analysis of AS\_PATH validation using real world BGP update messages through the SAPBGP.

## **1.2 Dissertation outline**

The remainder of this dissertation is organized as follows. In Chapter 2, we present survey on BGP, its security problems, current status of BGP research, and existing tools for preventing BGP problem. In addition, we discuss operation of BGP and how to prevent BGP hijacking.

In Chapter 3, we present architecture of BGPMAPS. For iBGP peers, BGPMAPS sends BGP update message including opaque extended community attribute. For eBGP peers, BGPMAPS notifies network administrators of invalid prefixes through alarm system. In addition, we analyze results of performance and experiments of BGPMAPS.

In Chapter 4, we present SAPBGP which monitors AS\_PATH attribute in BGP update messages whether each AS in the AS\_PATH attribute is connected to each other based on our policy database. We also present analysis of SAPBGP simulation using real world BGP update messages through the BGPmon project.

In Chapter 5, we analyze the results of BGPMAP and SAPBGP of performance test and experiments.

## **Chapter 2. Related Work**

This chapter provides an overview of how the Internet works and describes the current issues on BGP and appropriate solutions to deal with the BGP issues. In Section 2.1, an overview of BGP design and operation is presented, and then BGP's vulnerabilities and solutions are described in Section 2.2 and 2.3. In addition, past studies for BGP security are presented in Section 2.4. Furthermore, existing security tools for secure BGP are discussed in Section 2.5. We constructed BGP policy database to validate AS\_PATH attribute. We explained how to build policy-based database.

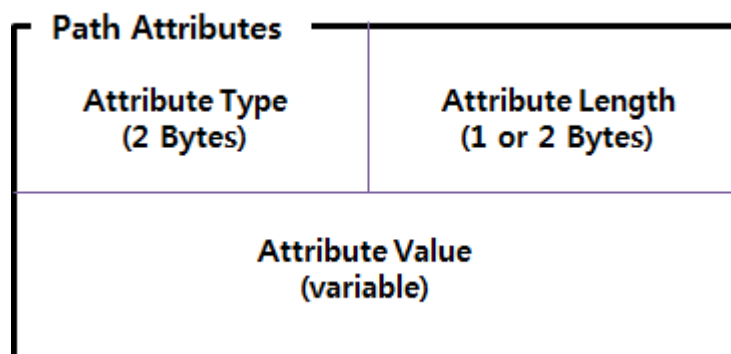
## 2.1 Design and Operation of BGP

Internet is a collection of Autonomous System (AS) that uses the BGP to exchange routing information about how to reach a certain block of IP addresses, called IP prefixes. Prefixes are expressed as 'prefix/length'. There should be at least one BGP router within an AS to exchange Network Layer Reachability Information (NLRI) with other ASes via UPDATE messages. In order to exchange UPDATE messages, both BGP routers should create a BGP connection by sending an OPEN message. Once two BGP routers, called BGP peers, are connected to each other, the BGP peers constantly exchange UPDATE messages to notify the peers of routing table changes that include the addition of new prefixes and the withdrawal of old prefixes. Each AS originates its prefixes and sends them to its peers, and the peers forward the prefixes to their neighbors.

A BGP message consists of OPEN, UPDATE, NOTIFICATION, and KEEPALIVE. An OPEN message is used to create a new BGP session. An UPDATE message is used to exchange routing table information with peers. A NOTIFICATION message is used to notify peers of errors while establishing peer relationships. Lastly, a KEEPALIVE message is used to check the availability of BGP peers. In order to create a BGP connection, one of the BGP routers sends an OPEN message to the other router. The OPEN message includes a version of BGP, AS number, hold time, BGP identifier, and optional parameters. The version of BGP indicates the current BGP version: 4 or 6. The value of source's AS number is assigned to the AS number. The hold time the

means maximum time in seconds that a router waits between a KEEPALIVE message and an UPDATE message. The BGP identifier is used to identify the source BGP router and the value of source IP address is assigned to the BGP identifier.

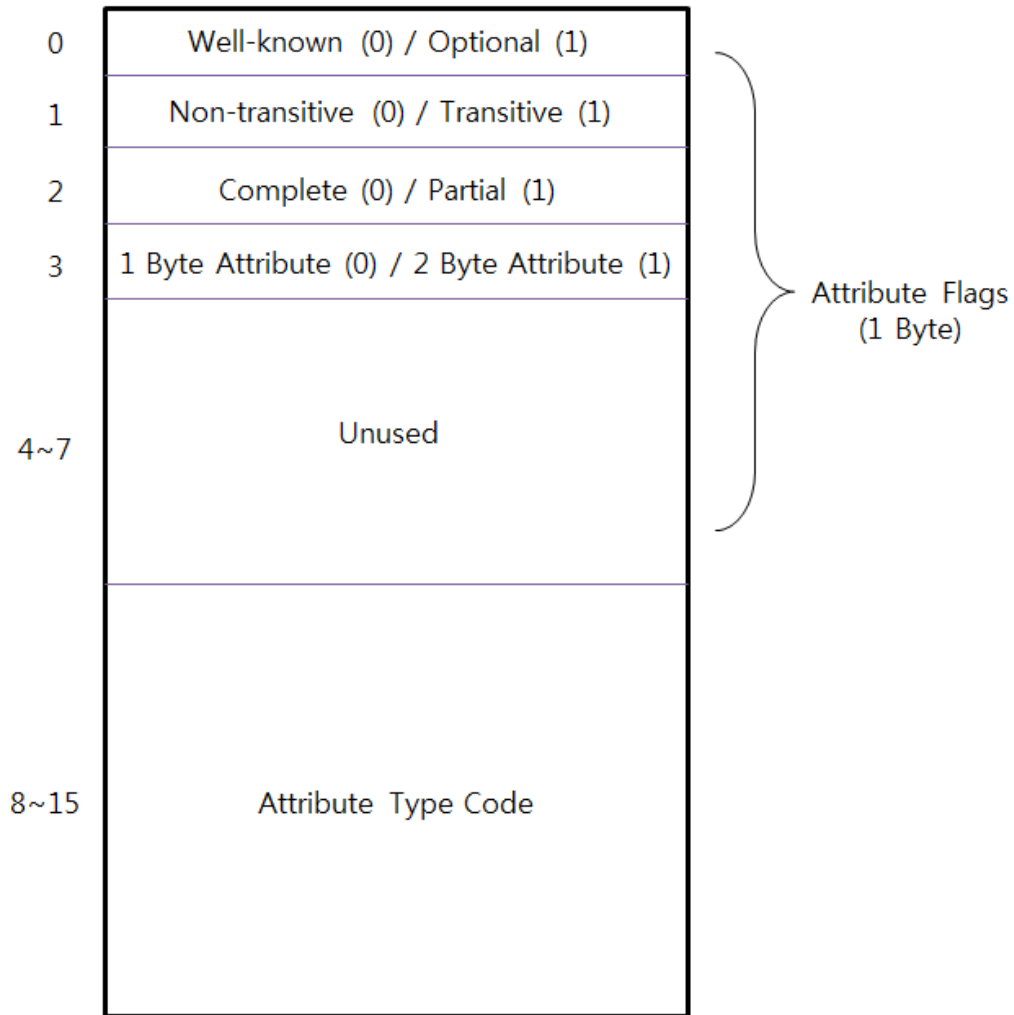
An update message includes the Withdrawn Routes, Path Attributes, and Network Layer Reachability Information (NLRI). Whenever new prefixes are added or removed, BGP routers communicate with neighbors using UPDATE messages. Withdrawn Route is used when the existing prefixes are removed, and NLRI is used when new prefixes are added. Multiple Path Attributes can be included in the UPDATE message. Path Attributes consists of three fields: attribute type, attribute length, and attribute value as shown in Figure 1.



**Figure 1. Path Attributes**

Attribute Type is two bytes and consists of Attribute Flags and Attribute Type Code as shown in Figure 2.





**Figure 2. Attribute Type (2 Bytes)**

The Attribute Type is used to define whether the attribute is an optional, well-known, transitive, or non-transitive bit. In addition, the value of the attribute type defines the attribute type, such as ORIGIN, AS\_PATH, NEXT\_HOP, MULTI\_EXIT\_DISC, etc. Table 1 enumerates the attribute types in details. For example, ORIGIN, AS\_PATH, and NEXT\_HOP attributes should be contained because they are referred to as well-known mandatory attribute indicated in the Attribute type flags. These Attribute Flags decides whether the attribute should be included or not. In addition, the size of

Attribute Length is determined according to Attribute Flag and the Attribute Length indicates the number of bits in which the Attribute Value can contain its value. Furthermore, Attribute Type contains Attribute Type Code indicated in the table below.

**Table 1. Path Attribute types**

Type Code	Attribute Name	Category	Source
1	ORIGIN	Well-known mandatory	RFC 4271
2	AS_PATH	Well-known mandatory	RFC 4271
3	NEXT_HOP	Well-known mandatory	RFC 4271
4	MULTI_EXIT_DISC (MED)	Optional nontransitive	RFC 4271
5	LOCAL_PREF	Well-known discretionary	RFC 4271
6	ATOMIC_AGGREGATE	Well-known discretionary	RFC 4271
7	AGGREGATOR	Optional transitive	RFC 4271
8	COMMUNITY	Optional transitive	RFC 1997
9	ORIGINATOR_ID	Optional nontransitive	RFC 1966
10	Cluster List	Optional nontransitive	RFC 1996
N/A	DPA	Destination point attribute for BGP	
N/A	Advertiser	BGP/IDRP route server	RFC 1863
N/A	RCID_PATH/CLUSTER_ID	BGP/IDRP route server	RFC 1863
14	Multiprotocol Reachable NLRI	Optional nontransitive	RFC 2283
15	Multiprotocol Unreachable NLRI	Optional nontransitive	RFC 2283
N/A	Extended Communities	N/A	RFC 4360
N/A	N/A	Reserved for development	

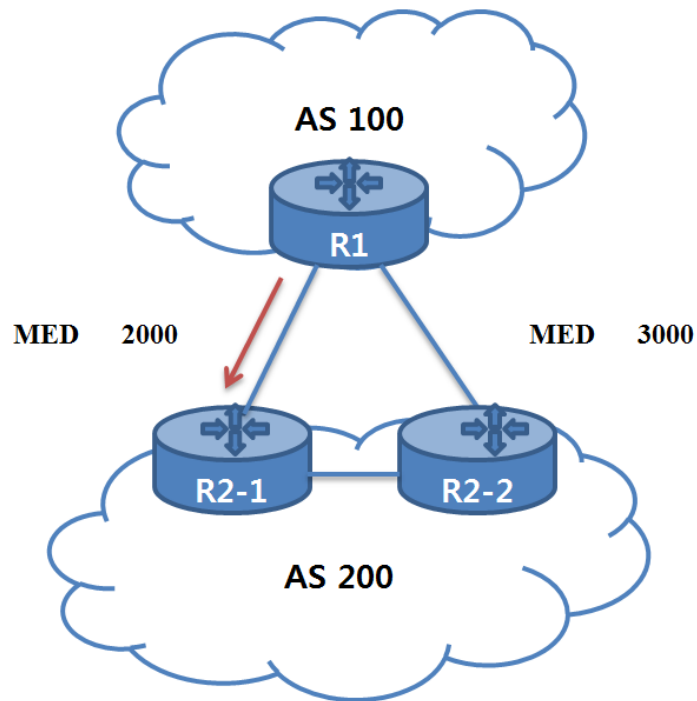
ORIGIN is a well-known mandatory attribute and describes how routes are introduced into the BGP path. If the routes are originated on a BGP router, then the value of ORIGIN is set to IGP. If the routes are originated from an EGP session, then the value of ORIGIN is set to EGP. If the routes are originated from normal router rather than a BGP router, such as redistribution from an IGP protocol (OSPF, RIP, or IS-IS), then the value of ORIGIN is set to Incomplete [26], [27], [28], [29].

AS\_PATH is a well-known mandatory attribute and the main purpose of AS\_PATH to prevent routing loops. A BGP router can know whether an UPDATE message has passed or not after checking the UPDATE message. If its own AS number is included in the AS\_PATH attribute, then the BGP router can ignore the UPDATE message. Whenever a BGP router receives an UPDATE message, the BGP router adds its own AS number to the AS\_PATH attribute [7].

NEXT\_HOP is a well-known mandatory attribute and indicates the IP address of the last eBGP router. If an UPDATE message is forwarded within the same AS, then the value of the NEXT\_HOP attribute is not changed. NEXT\_HOP is not necessarily the physical next hop. NEXT\_HOP is used to indicate the IP address of the next-hop BGP router when IP packets need to be transferred to the destination [7].

MULTI\_EXIT\_DISC is a non-transitive optional attribute and used in the process of selecting best path when a BGP router has more than two BGP connections with the same AS. In Figure 3, R2-1 sends MED value 2000 to R1

in AS 200 and R2-2 sends MED 3000 to R1. Then, R1 will prefer the lower metric and send all Internet traffic for AS 200 through R2-1 [7].



**Figure 3. MED example**

When BGP routers have more than two routes, the best path is decided by comparing various BGP attributes, which is called a tie-breaker. The tie-breaker procedures are as follows:

- The highest weight is preferred
- The highest local preference is preferred
- Locally originated routes are preferred
- The shortest AS\_PATH is preferred
- (The lowest origin code is preferred)
- The lowest MED is preferred
- EBGp path is preferred

- The lowest IGP metric to next hop is preferred
- The oldest route for EBGp path is preferred
- The lowest neighbor BGP router ID is preferred

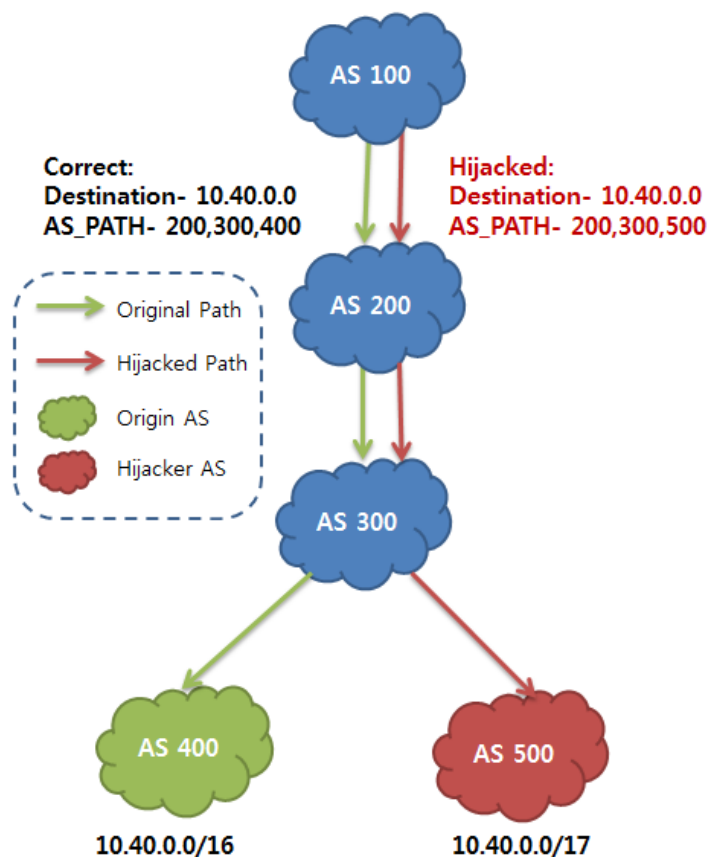
## **2.2 BGP's Vulnerabilities**

BGP is used to find the best path to reach the destination between the source AS and the destination AS. In selecting the best path, the length of prefix and the number of hops are considered. Hijackers use those two characteristics of BGP to illegally draw Internet traffic to their AS. First, a longer prefix has a higher priority. AS administrators can announce any prefixes, which means the AS administrator intentionally/unintentionally can announce others' prefixes, and it changes the destination of Internet traffic. Secondly, a shorter path has a higher priority. When a BGP update message is forwarded among ASes, each AS's ASN is added to the AS\_PATH attribute. A hijacker can manipulate the AS\_PATH attribute to change AS paths of the Internet package. In addition, hijackers can pretend their ASes are connected to other ASes, by manipulating the AS\_PATH attribute in the BGP message, even though their ASes are actually not connected to each other. Therefore, when the best path is selected, illegal changes of AS\_PATH attribute influence the process of the best path selection.

### **2.2.1 IP hijacking**

Once BGP routers are connected to each other, the BGP routers fully trust other routers. If a BGP router intentionally originates a bogus prefix to neighbors, the neighbors that receive the announcements trust the prefix and their traffic is hijacked by the hijacking router.

Figure 4 shows a scenario of IP hijacking. AS 500 is trying to hijack the Internet traffic heading for AS 400. AS 400 announces 10.40.0.0/16 to neighbors and traffic in AS 100 is going to 10.40.0.0. However, if AS 500 announces a bogus prefix, 10.40.0.0/17, to AS 100, then the traffic in AS 100 goes to AS 500 because 10.40.0.0/17 is more specific than 10.40.0.0/16. As a result, AS 100 takes the 10.40.0.0/17 as the destination.



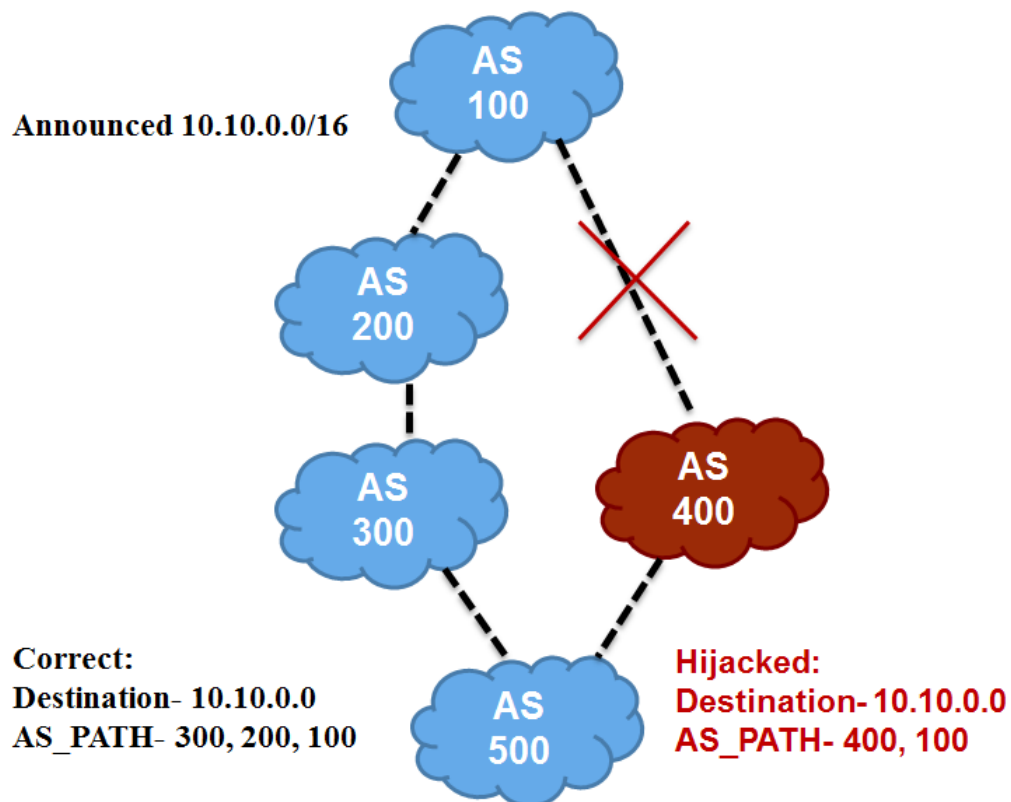
**Figure 4. IP prefix hijacking**

### 2.2.2 AS path hijacking

AS path hijacking is the most severe problem that happens in BGP because it is hard to be detected [30]. AS path hijacking not only changes routes of Internet packets, but also sends the Internet packages to the right destination, which means victims of AS path hijacking hardly realize that their Internet packets are monitored or manipulated by AS path hijackers.

Nowadays, there are many unknown BGP attacks [13], [14] because victims of the hijacking cannot notice any changes except latency which is caused by the hijacker because the Internet packets traverse more AS hops.

A BGP router inserts its own ASN into the AS\_PATH attribute in update messages when the BGP router receives the update message from neighbors. However, the BGP router can insert one or more ASNs into the AS\_PATH attribute in update messages other than its own ASN. In addition, a BGP router might pretend as if the BGP router is connected to a certain BGP router by manipulating data contained in BGP updates. Figure 5 demonstrates a scenario of manipulating BGP update messages.



**Figure 5. Manipulating AS\_PATH attributes**

Suppose AS 400 has a connection to AS 500 and creates a fake BGP announcement to pretend that AS 400 received a BGP message originated by AS 100 and forwarded the update message to AS 500 even though AS 100 and AS 400 actually don't have a BGP connection. In terms of AS 500, the traffic heading for prefix 10.10.0.0/16 will choose AS 400 as the best path because AS 500 selects the shortest path and AS 400 is shorter than AS 300. Even if the AS 500 can conduct origin validation, the AS 500 cannot prevent this attack because prefix and ASN information is correct. As a result, AS 400 will have the traffic heading for prefix 10.10.0.0 and might start another attack using the traffic, such as a Man-In-The-Middle (MITM) attack.



### **2.2.3 Peer spoofing**

Spoofing is the most common attack in network protocol. In BGP, spoofing indicates that a hijacking router creates a new BGP update as if the BGP message is originated from other BGP routers than its own source. By spoofing attack, the hijacking router can insert false information to a BGP peer's routing tables.

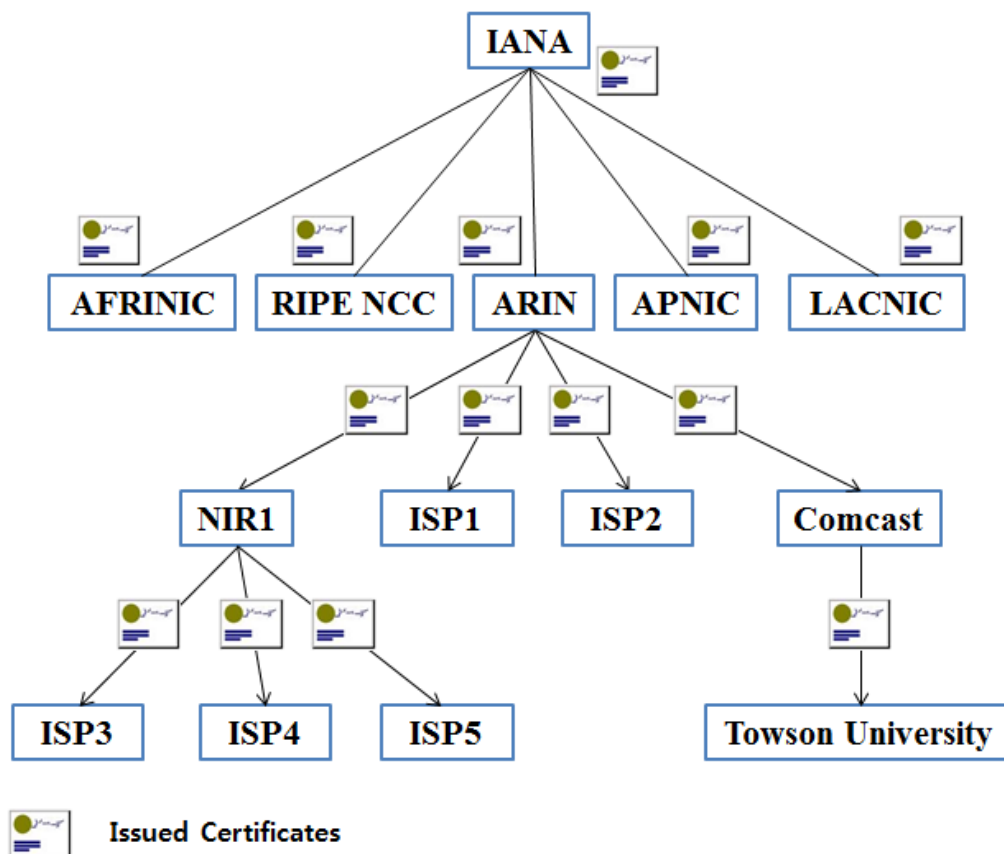
## **2.3 BGP information validation**

In order to validate BGP update message, origin information of a BGP update message needs to be checked whether authorized BGP router originated its prefixes or not, which is called origin validation. In addition, AS-PATH information in a BGP update message needs to be checked whether AS-PATH attribute has been changed or not, which is called path validation.

### **2.3.1 Origin validation**

An origin validation means to verify whether the originator of update message has been authorized to announce its prefixes. In order to validate originators, the Resource Public Key Infrastructure (RPKI) was implemented by SIDR working group on January in 2013 and is currently used for origin validation. RPKI is a Public Key Infrastructure (PKI) [31], [32] where an organization called IANA manages officially verifiable Internet resources that are the allocation of hierarchy of IP addresses, Autonomous System Numbers (ASN), and signed objects for routing security. IANA is the trust anchor that

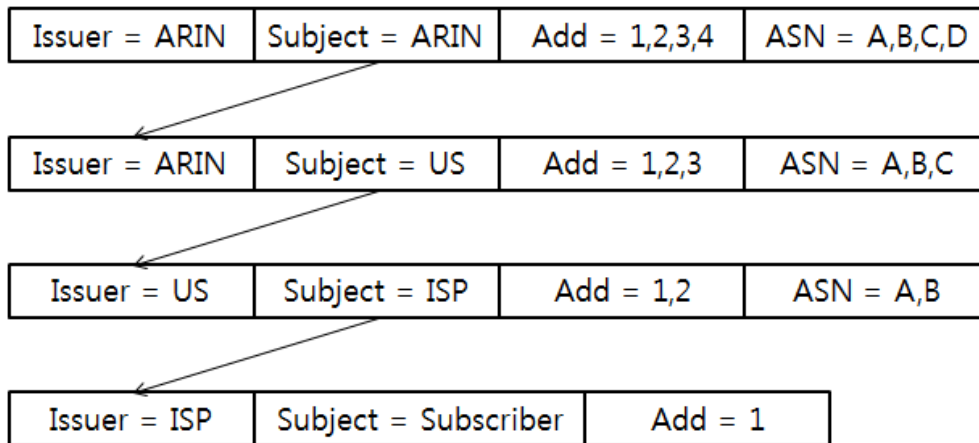
allows third party to officially validate assertions according to resource allocations. The authorization is hierarchically assigned from IANA to the Regional Internet Registries (RIRs), Local Internet Registries (LIRs), National Internet Registries (NIRs), and Internet Service Providers (ISPs) as shown in Figure 6.



**Figure 6. Hierarchy of the RPKI**

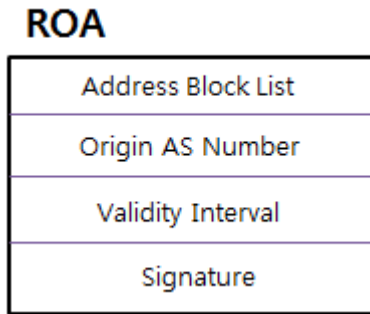
There are five RIRs and they act as trust anchors like IANA. The RIR issues certificates to NIR, ISP and subscribers. NIR and ISP are allowed to issue certificates to downstream providers and to subscribers. IP address holders specify which ASes are authorized to announce their own IP address

prefixes.



**Figure 7. Certificate Chain**

Figure 7 explains how a subscriber hierarchically gets certificates regarding their IP address. For example, ARIN issues certificates for US regarding addresses 1, 2, and 3 and ASN A, B, and C as shown in Figure 7. US issues certificates to ISP regarding address 1 and 2 and ASN A and B. Then, a subscriber can get a certificate from ISP regarding its IP addresses. As shown in Figure 8, the certificate, called Route Origin Authorizations (ROAs) [33] is a digital object formatted following the Cryptographic Message Syntax Specification (CMS) [34] [35] and composes of origin AS Number, validity date range, and one or more IP addresses with a CIDR block. If the address space holder needs to authorize multiple ASes and the IP prefixes are the same, the holder should issue multiple ROAs.

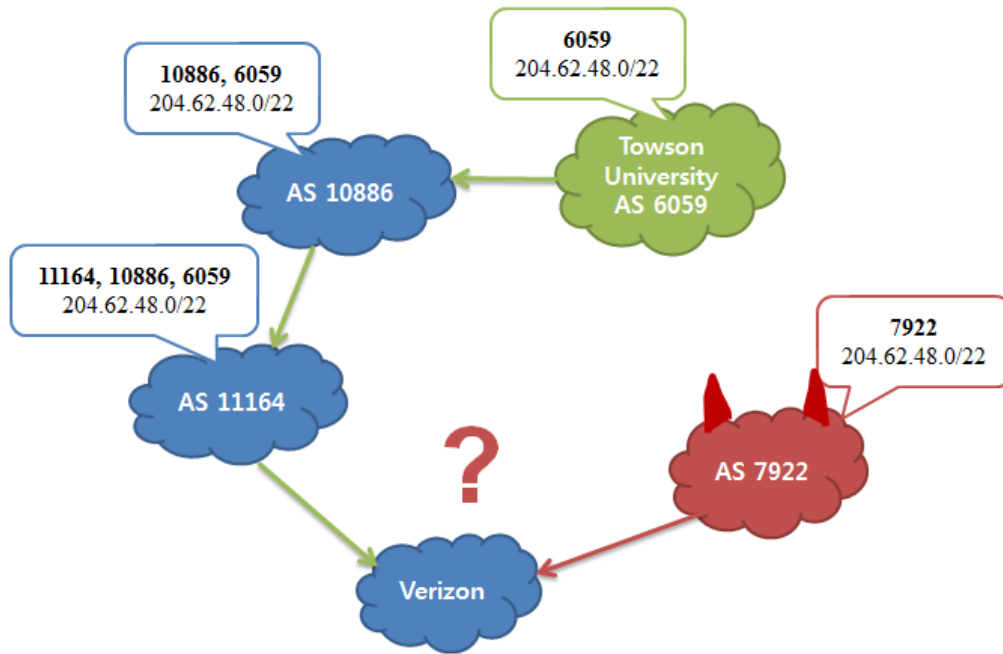


**Figure 8. ROA Format**

The value of Address Block List is more than one prefix, corresponding to the NLRI that the ROA signer authorizes for prefix announcements by one or more ISPs. The value of origin AS number that is authorized to announce the prefixes indicated in the address block list. Validity interval indicates the start and end date for which the ROA is valid. Signature includes pairs of information that is used to verify the ROA. One is certificate pointer that directs its parent so that the certificate has been issued by CA. The other one is signature that is digitally signed hash data including address block list, origin as number, validity interval, hash algorithm, and digital signature algorithm. Therefore, if a prefix hijacker announces other's prefixes, other network operators can check whether the announcement is invalid after comparing the IP prefixes and ASN which are included in the update message to the ROA.

For example, as shown in Figure 9, there are five ASes. Towson University (AS 6059) announced its prefix 204.62.48.0/22. As the update message is transferred, each ASN is added to the AS\_PATH attribute, and finally Verizon receives the update message and knows how to reach the prefix

204.62.48.0/22 through the AS\_PATH attribute. However, if a hijacker sends the same prefix 204.62.48.0/22, then Verizon will choose AS 7922 as the final destination because the number of hops is shorter than the other as shown in Figure 8.



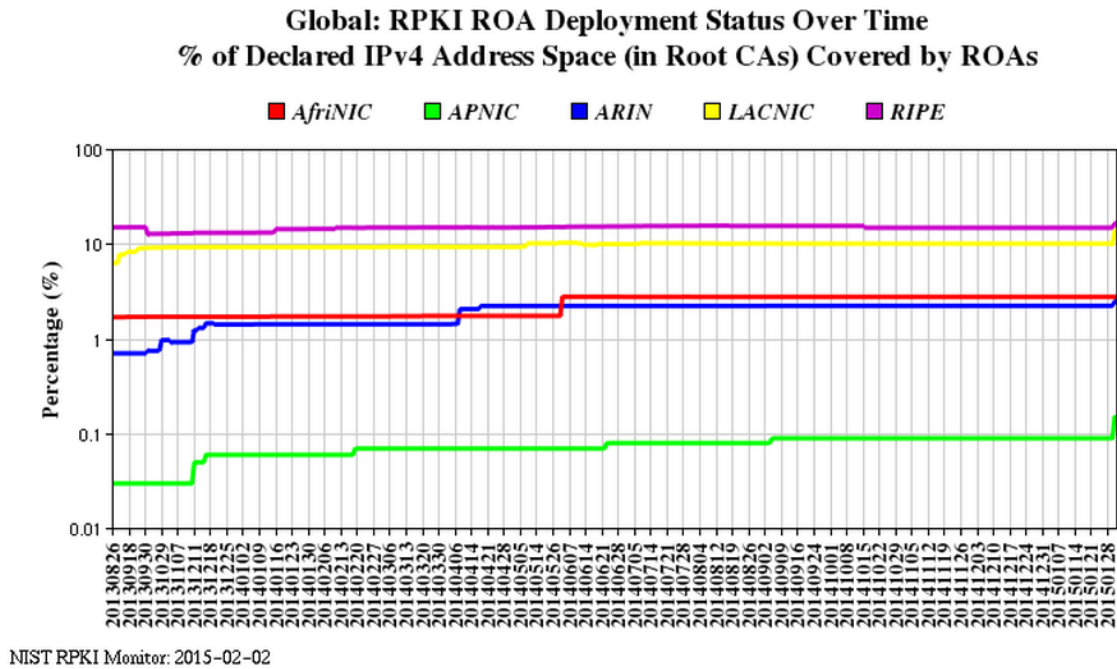
**Figure 9. Scenario of IP hijacking**

At this moment, if Verizon maintains ROAs and checks the ROAs then Verizon will realize that AS 7922 is not authorized to originate the prefix 204.62.48.0/22 because the ROA as shown in Table 2, indicates that AS 6059 has been authorized to announce the prefix 204.62.48.0/22.

**Table 2. AS 6059's ROA**

ROA
204.62.48.0/22
AS 6059

According to the NIST RPKI report issued on the second of February 2015 as shown in Figure 10 [36], the small number of address spaces is covered by ROAs.



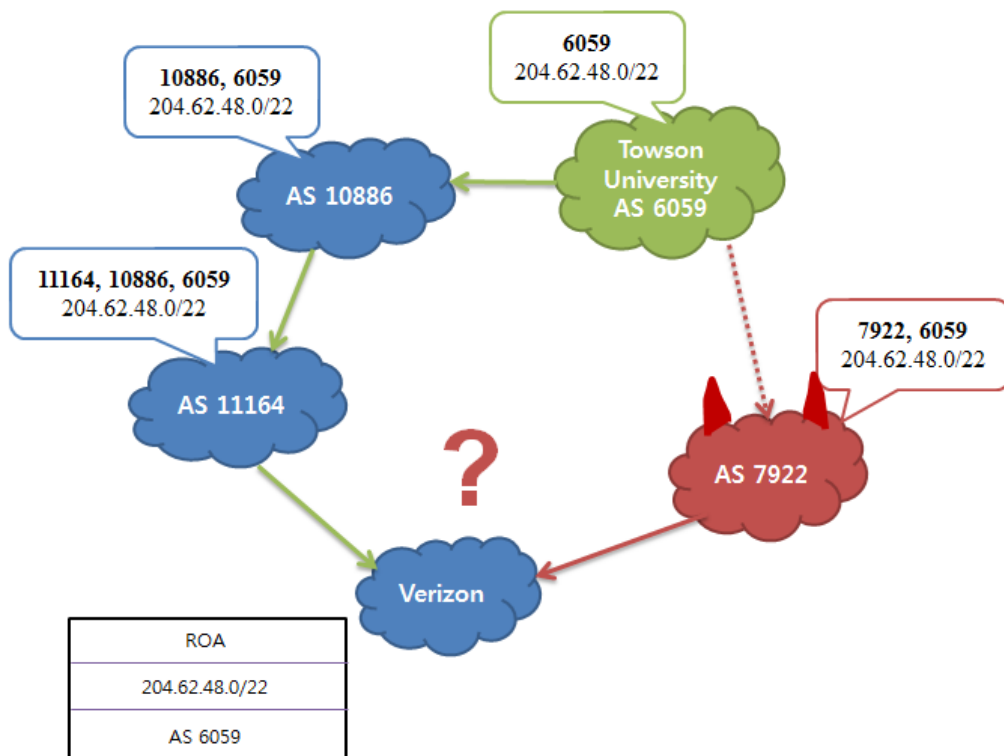
**Figure 10. IPv4 covered by ROAs**

If all address spaces are authorized by its address holders, then IP prefix hijacking will be fully prevented by RPKI.

### 2.3.2 Path validation

IP hijacking can be completely prevented by RPKI if every address is covered by the ROAs. However, even though all of the IP addresses are covered by the ROAs, hijackers can try an AS-PATH hijacking by changing the AS\_PATH attribute in the update message. In other words, the origin validation cannot assure that the update message has been originated by the authorized BGP router. Figure 11 shows a scenario of the AS-PATH hijacking.

Even though Verizon has the ROA for the prefix 204.62.48.0/22, Verizon cannot check that the UPDATE message has been originated from AS 6059 because the origin validation doesn't guarantee that the update message has not been modified by malicious routers. In Figure 11, AS 7922 pretends as if AS 7922 has the BGP connection to AS 6059 as well as pretends to receive UPDATE message from AS 6059 and to forward the UPDATE message to Verizon. Then, in the perspective of Verizon, the shortest path to reach 204.62.48.0/22 is AS 7922. As a result, it is important for each BGP router to check that the UPDATE message has not been changed on its way.

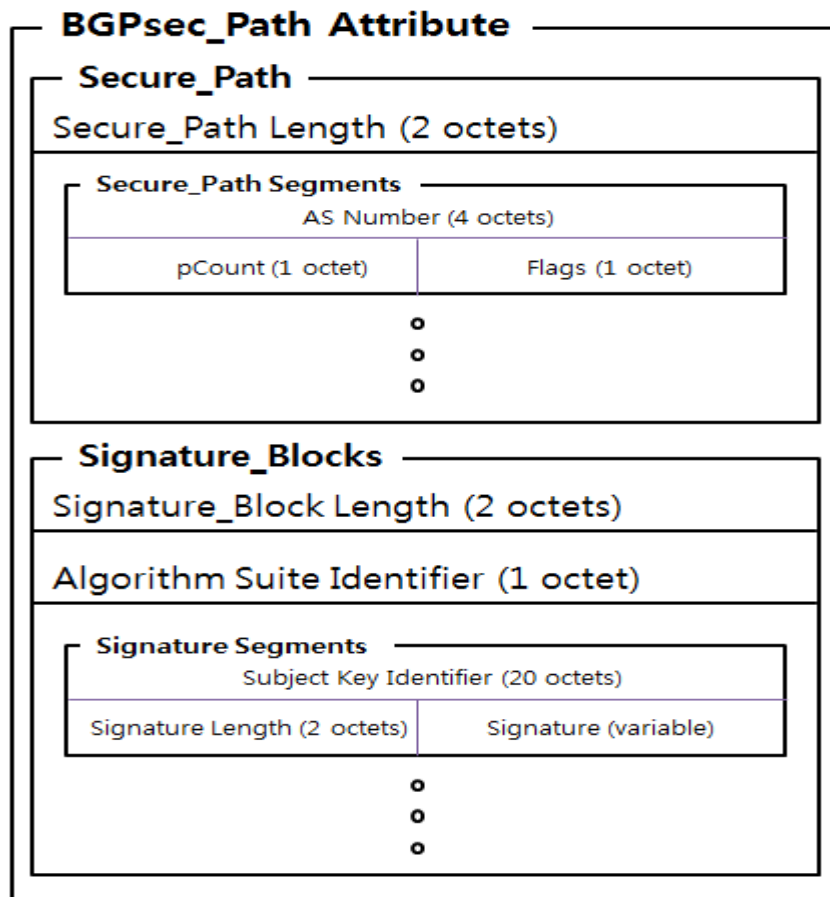


**Figure 11. 1-hop hijacking**

In order to prevent the AS-PATH hijacking, BGP routers should check whether an incoming update message is changed or not. In addition, the BGP routers check whether the sequence of ASes in the AS\_PATH attribute is the

same as the actual propagation path of the BGP update message. Currently, a SIDR working group is designing BGPsec to cryptographically prevent the AS-PATH hijacking. In BGPsec, an optional and non-transitive path attribute, BGPsec\_Path attribute, is included in BGP update messages. BGPsec depends on RPKI certificates and a BGP router that wants to send BGP update messages that includes the BGPsec\_Path should have a private key associated with the BGP router's AS number. When the BGP router originates IP prefixes, the BGP router signs the update message with its private key so that any BGP router that receives the update message can check that the update message has been originated by the right BGP router by verifying the signature with the public key corresponding to the private key. In addition, BGP routers who receive the BGP update message sign the BGP update message with their private key and forward the BGP update message to neighbors. If every router that receives and forwards the BGP update messages signs the BGP update message, the BGP update message can be considered as the message that has not been illegally changed by hijackers.





**Figure 12. BGPsec\_Path Attribute**

In order to protect BGP update message, especially to protect AS\_PATH attributes, the BGP update message should carry the secured information such as digital signature. We call the BGP update message including a BGPsec\_Path attribute BGPsec update messages as shown in Figure 12. The AS\_PATH attribute in BGP update messages is replaced with BGPsec\_Path attribute in the BGPsec update messages. The BGPsec\_Path attribute contains a Secure\_Path attribute and sequence of one or two Signature\_Blocks. Basically, the BGPsec\_Path attribute is logically equivalent to the AS\_PATH attribute, but the BGPsec\_Path attribute includes signature

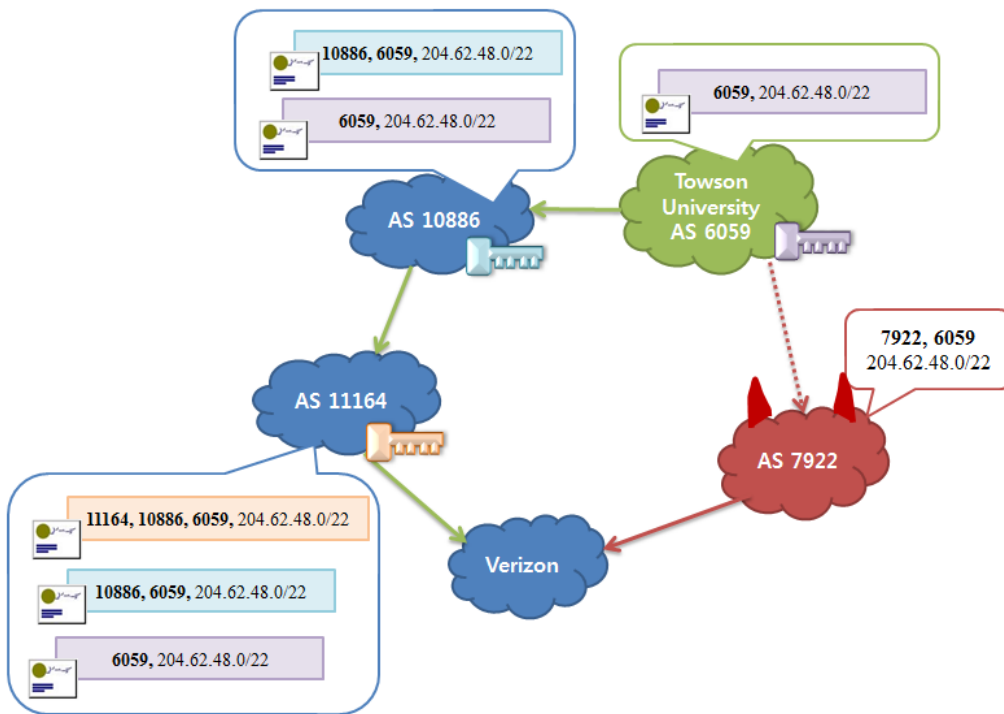
blocks for security methods. The Secure\_Path attribute includes Secure\_Path length and one or more Secure\_path segments. The Secure\_path segments consist of AS number, pCount, and Flags. The value of AS number is the AS number of the BGP router that originates or forwards BGP update messages. The value of a pCount attribute is the number of repetitions of an AS number that the signature will cover. According to the value of pCount, BGPsec routers forwards the BGPsec update messages without generating multiple signatures. For example, as shown in Figure 13, if the value of pCount is 3 for AS 11164, the BGPsec update message is forwarded to peers without creating a signature.

<b>Before applying pCount</b>						
AS-PATH:	6059	10886	11164	11164	11164	7922 7922
<b>After applying pCount</b>						
pCount:	1	1	3	2		
AS-PATH:	6059	10886	11164	7922		

**Figure 13. Example of the pCount attribute**

The value of a Flags attribute is the Confed\_Segment flag that is the first bit of the Flags attribute that contains eight bits and is used to indicate that BGPsec router that created the current BGPsec update message is currently sending to a BGPsec router within the same Autonomous System. In that case, the value of Confed\_Segment flag is set to 1 and in all other case, the flag is set to 0. The remaining seven bits of the Flags attribute are currently unused and are set to 0. According to the Secure\_Path attribute, one or more

Signature\_Block attributes are required. A Signature\_Block contains Signature\_Block Length, Algorithm Suite Identifier, and Sequence of Signature Segments. The Signature\_Block Length attribute indicates the total number of octets in Signature\_Block. The Algorithm Suite Identifier attribute specifies a digital signature algorithm used to create a digital signature and a digest algorithm for cryptographic hash function producing a hash value. The Signature Segment attribute consists of Subject Key Identifier, Signature Length, and Signature. The Subject Key Identifier attribute includes the Subject Key Identifier extension of the RPKI router certificate that is used to verify signatures. The Signature Length attribute field contains the size of the Signature attribute in the Signature Segment. The Signature attribute includes a digital signature to protect the BGPsec\_Path attribute.



**Figure 14. Protecting the 1-hop hijacking by BGPsec**

Figure 14 depicts how the BGPsec update message works to protect the 1-hop hijacking. As already shown in Figure 11, Verizon cannot protect the 1-hop hijacking, even though Verizon can conduct origin validation. In order to prevent 1-hop hijacking, every BGPsec router needs to use a BGPsec update message instead of a BGP update message and sign the BGPsec update message with its private key either when the BGPsec router originates or when the BGPsec router forwards it to neighbors. When a BGPsec router receives a BGPsec update message, the BGPsec router needs to conduct following preprocessing:

- Check that each Signature\_Block attribute includes one Signature segment for each Secure\_Path segment.
- Check that the BGPsec update message does not contain an AS\_Path attribute
- Check that Secure\_Path segments do not contain a Flags attribute that is set to one if the BGPsec message sender is not a member of the BGPsec router's confederation.
- Check the pCount attribute to see if the value is set to zero even though the value is not supposed to be set to zero

If any one of the above procedures fails, then the BGPsec\_Path attribute is considered as malformed. After checking preprocessing, BGPsec routers conduct the following procedures to examine the Signature\_Blocks in the BGPsec\_Path attribute.

- Verify the signatures with its public key that was obtained from the valid RPKI data and check the data in the BGPsec\_Path attribute whether AS, SKI, and Public key are correct or not
- Conduct the digest algorithm with the given algorithm on data such as AS Number of Target AS, Origin AS Number, pCount, Flag, Algorithm Suite Identification, NLRI Length, and NLRI prefixes
- Verify the signature with the signature validation algorithm, which are the public key, the value of the Signature attribute, and the digest value

If at least one Signature\_Block attribute for a BGPsec\_Path attribute is valid, the BGPsec update message is considered as valid, otherwise the BGPsec update message is considered as invalid.

## **2.4 BGP Security Architectures**

In this section, we discuss existing six approaches to BGP security: S-BGP [37], SO-BGP [38], psBGP [39], IRV [40], pgBGP [41], and SPV [42].

### **2.4.1 Secure BGP(S-BGP) and Secure Origin BGP(SO-BGP)**

S-BGP has been considered the most complete solution to project BGP security [37]. The main purpose of S-BGP is to protect a BGP update message from modification and to prevent a BGP router from accepting fictitious NLRI by validating update messages through Resource Public Key Infrastructure (RPKI) [15]. RPKI has two types of attestations.

One attestation is a digitally signed statement for S-BGP router to have the right to origin NLRI to other BGP routers. The S-BGP router sends an update message which includes the attestation digitally signed with its private key to neighbors. Once the neighbors take the update message including attestation, the neighbors check its attestation to verify if the message has been changed while it is being transmitted.

The other attestation is a statement for path validation. If a S-BGP router receives its neighbor's update message, the router can verify that the update message has been traversed in the right sequence of ASes according to AS\_PATH, which is included in the update message.

SO-BGP is similar to S-BGP in using PKI, but SO-BGP uses the concept of a web of trust rather than the hierarchical PKI that S-BGP uses [38]. In SO-BGP, an EntityCert is used to send and receive keys between S-BGP routers. The EntityCert ties an AS number to a public key relevant to a private key the AS will sign other SO-BGP router's certificates.

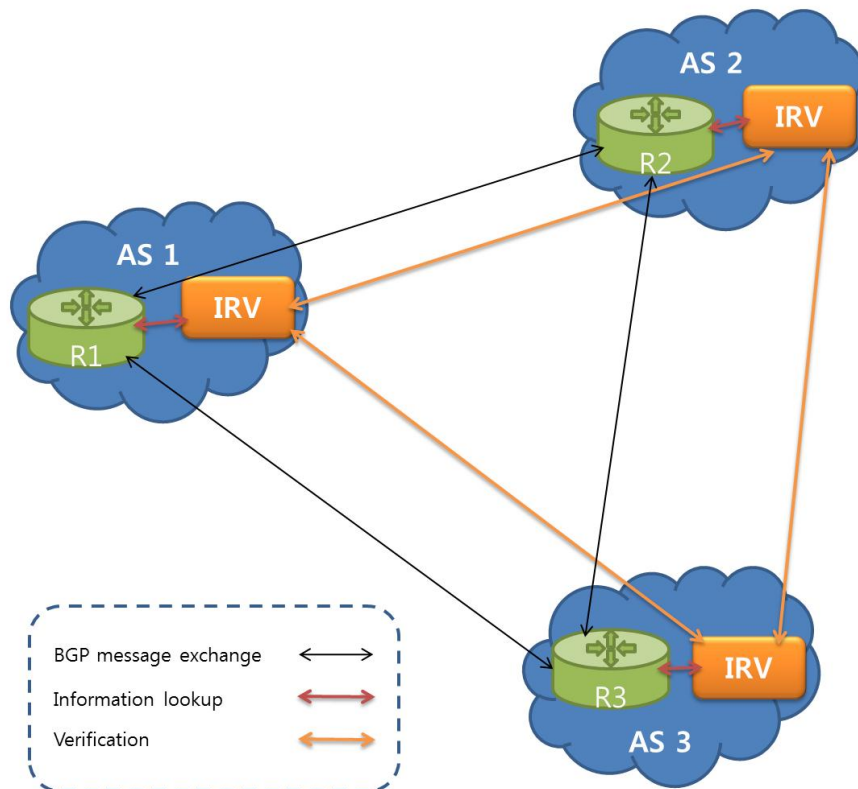
#### **2.4.2 Pretty Secure BGP (psBGP)**

psBGP combines the best features from S-BGP and SO-BGP. First, psBGP uses a centralized trust model to validate AS number by obtaining a public key certificate from a number of certificate authorities, such as RIRs, and by binding an AS number to a public key [39]. In order to validate IP prefix ownership, psBGP uses a decentralized trust model. Each AS creates a Prefix Assertion List (PAL), which is composed of IP prefix ownership

assertions of the local ASes and its peering ASes. The PALs of peers are checked for consistency whenever a new origin is validated.

### 2.4.3 Inter-domain Route Validation (IRV)

The main concept of IRV is that each AS designates an IRV system through which BGP router validates an incoming update message [40]. Figure 15 shows a simple topology, which is composed of BGP routers and IRV systems. Each designated IRV system communicates with a BGP router. In addition, the IRV system processes a request from a remote IRV for the path validation.



**Figure 15. Inter-domain Route Validation**

Most existing BGP security solutions such as S-BGP validate BGP data by using digitally signed statement. However, IRV server in each AS receives query from the BGP router to validate whether an update message includes correct information. As a result of this, the IRV model can reduce encryption/decryption computation time. However, this IRV model has several problems that are unclearly specified. For example, it ambiguously specifies how an IRV response is validated.

#### **2.4.4 Pretty Good BGP (pgBGP)**

The main feature of pgBGP is a capability to recognize whether an incoming update message is valid or not by using its historical routing data [41]. In order to do this, pgBGP routers need to exercise a certain amount of routes that they adopt into their routing table. Through the routing table, pgBGP prevents IP prefix hijacking attacks and misconfigurations. If a pgBGP router receives an update message, which includes new origins, the pgBGP router considers the update message as an anomalous origin, and the pgBGP ignores the update message including the origin unless the origin is in its database. In order to add a new origin, if the origin remains in the router after 24 hours, the origin is added to the normal database. If an origin has not been seen in the routing table for a long time, the origin is removed from the database.

#### **2.4.5 Secure Path Vector (SPV)**



SPV uses a symmetric-key cryptography method for preventing update messages from being modified by a malicious router [42]. SPV has three characteristics. Firstly, SPV contains private keys inside the update message itself. Secondly, SPV doesn't validate the AS that inserts itself onto the path. Finally, SPV offers BGP security without computation time. As a result of these characteristics, SPV is even faster than S-BGP.

However, SPV is vulnerable to some attacks. For example, a malicious router can insert fake ASNs between its two ASNs. Furthermore, the probability of truncation is high when an SPV router receives several update messages from a single prefix.

## **2.5 Existing security tools**

BGP-SRx, developed by the National Institute of Standards and Technology (NIST), consists of the SRx Server, the SRx API, and the Quagga SRx [43]. SRx provides a proxy with APIs, which allows the proxy to be embedded on the router and communicate with the SRx Server. The Quagga SRx is a software router on which the proxy is embedded. The SRx Server is connected to the RPKI validation cache, so the SRx Server can validate BGP announcements by comparing the BGP announcements to ROAs in the RPKI validation cache.

Prefix Hijack Alert System (PHAS) is a system that detects an attempt to hijack prefixes, owned by other BGP routers, with BGP routing data collected by BGP collectors. It also notifies prefix owners of the hijack attempt

through a reliable manner. However, PHAS does not guarantee to detect anomaly advertisements [18].

BGPmon is a monitoring infrastructure, implemented by Colorado State University that collects BGP messages from various routers that are distributed and offers the BGP messages as the routes for destinations are changed in real-time [24]. Any BGP router can be a source that offers real-time update messages if the BGP router is connected to BGPmon. Currently, 9 organizations participate in the BGPmon project as a source router. In addition, BGPmon collects Multi-threaded Routing Toolkit (MRT) format [44] live streams from the RouteViews project through indirect peering. The MRT format defines a way to exchange and export routing information through which researchers can be provided BGP messages from any routers to analyze routing information. Clients can be connected to the BGPmon via telnet and receive the live BGP stream in real time.

Cyclops is a system that collects real-time updates of hundreds of routers and displays a graphic view of how the routers are connected to each other [45]. As a result, network administrators can use the tool to detect and diagnose BGP misconfigurations or BGP hijacking.

Argus is an agile system that receives real-time updates from BGPmon and daily updates from Center for Applied Internet Data Analysis (CAIDA) iPlane. Based on the updates, Argus checks whether the update has an anomalous origin according to its local routing information database [17].

After finding the suspicious prefixes, Argus makes a final decision by computing the fingerprint for the suspicious prefixes.

## **2.6 BGP Policy Database**

We use BGP policy information to conduct the BGP path validation. In collecting BGP policy information, we used the RIPE data repository provided by RIPE NCC. The RIPE data repository is available for anyone that needs BGP information. The original purpose of the BGP policy is to filter incoming BGP messages and to choose BGP peers that will receive the BGP messages using BGP import and export policies. BGP router operators voluntarily upload their BGP policies to Internet Route Registries (IRR) through a predefined format, called Routing Policy Specification Language (RPSL) [46] that is provided by IRR. RIPE NCC database has been part of IRR and is composed of a set of online databases that is available for research purposes. In addition, RIPE NCC monitors Internet routing data and stores links between the routing data that has been seen by RIPE NCC. RIPE NCC provides users with RIPE Data Repository that contains BGP peer information. Through this information, we can know if any ASes are connected to other ASes. This peer information has been collected by either Routing Information Service (RIS) or IRR. RIS has collected and stored Internet routing data from several locations all over the world since 2001.

# **Chapter 3. The BGPMAPS**

## **3.1 Overview of the BGPMAPS**

Over the past decades, the number of BGP hijacking incidents gradually increased. Even enormous and high-tech companies, such as YouTube, Google, and so on became one of the victims by IP hijacking attacks. In order to protect from IP hijacking, many studies were conducted and several validation tools were implemented, such as BGPmon, PHAS, and Argus. Those validation tools keep collecting BGP update messages and keep validating the BGP update messages from the real world. Every time the malicious prefixes are found, those tools could only warn network administrators about the hijacking. However, it is important for the administrators to quickly block the bogus prefixes because thousands of Internet packets will be transferred to the wrong destination within a very short moment. In addition, the administrators manually block the malicious prefixes using a keyboard. As the number of ASes and prefixes has increased steadily over the past years, the manual configuration would be the problem because the administrators need to keep their eyes on the routing table all the time and the size of routing table has gradually increased over time. In order to solve those problems, the BGPMAPS notifies iBGP peers of the malicious prefixes using BGP update messages so that the iBGP peers can block the malicious prefixes automatically if the iBGP peers can handle the opaque extended community. The opaque extended community is one of the BGP attributes included in the BGP update message and currently a work in progress by the SIDR working group. The opaque extended community is used to carry the

validation state within the same AS to influence the decision process of BGP routes.

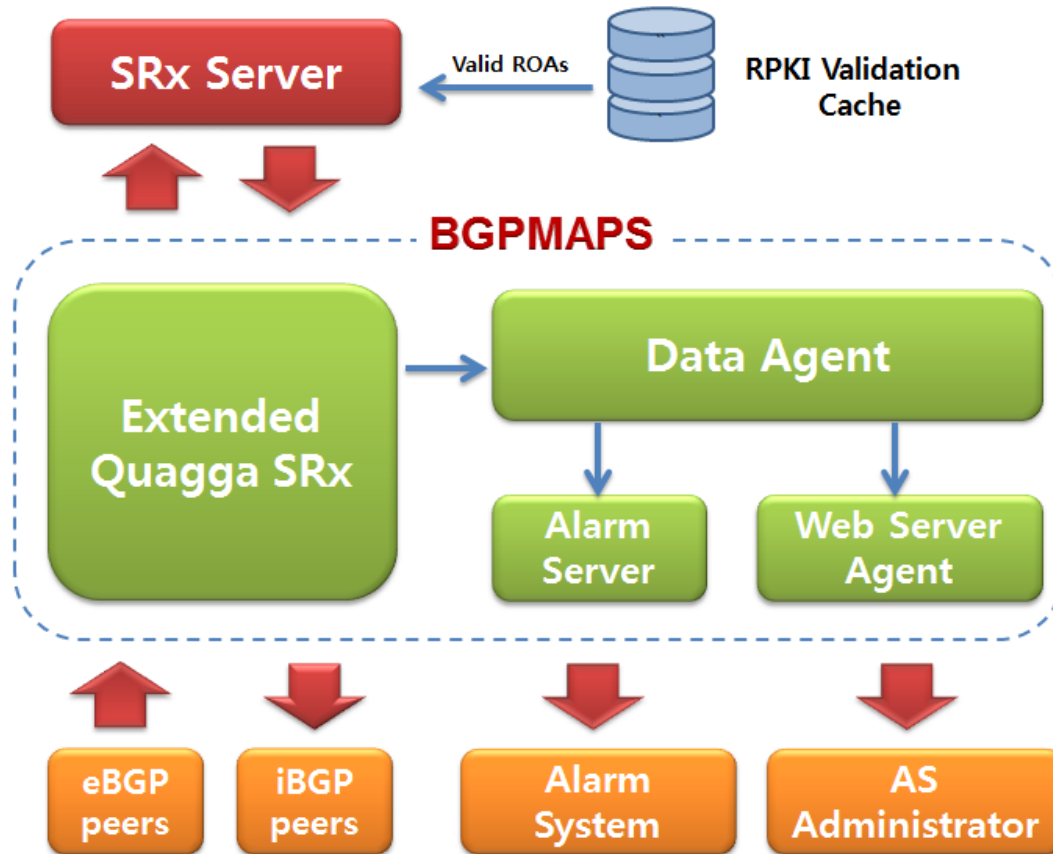
### **3.2 Architecture of the BGPMAPS**

Our beginning approach to prevent IP hijacking was to send BGP update messages including withdrawn routes instead of sending BGP update message including an opaque extended community attribute when an invalid message is detected. However, the BGP update message including withdrawn routes should be sent by only the BGP router that announced BGP update messages [8]. Therefore, the withdrawn routes cannot be used to notify neighbors of the invalid prefix. On the other hand, the BGP update message including opaque extended community attribute can be used even though the opaque extended community is currently a work in progress by SIDR working group [47].

The BGPMAPS consists of the Extended Quagga SRx (ex-Quagga-SRx), Data Agent (DA), Alarm Server, and Web Service Agent (WSA). The ex-Quagga-SRx receives update messages through BGP connection. Then, the ex-Quagga-SRx sends update message information such as ASN, prefix, and max length to the SRx Server.

The SRx compares the update message information to the ROAs and returns the result of validation to the ex-Quagga-SRx. The Data Agent receives the result of the BGP update message from the ex-BGP-SRx. If the result is invalid, the Alarm Server notifies the Alarm System of the invalid update

message. Then, the Alarm System makes sounds of warning, and BGP router administrators can check the invalid update message through the web interface.



**Figure 16. The architecture of the BGPMAPS**

### 3.3 Operational requirements

In the design of the BGPMAPS, we consider three requirements: (1) it should store a large amount of prefixes' history, (2) it should detect the bogus prefix and notify iBGP peers of the bogus prefix quickly and automatically when a bogus prefix is forwarded from BGP peers of other ASes, and (3) it should be a long living process.

In the past decade, network administrators were worrying about the growth of the number of routes in the Internet because network routers have a limited memory capacity of storing many prefixes in the routing table. The router may need an extra space to save additional routing information for preventing IP hijacking. As the number of prefixes is increased, the results of the validation are saved in the local database instead of the main memory of the router.

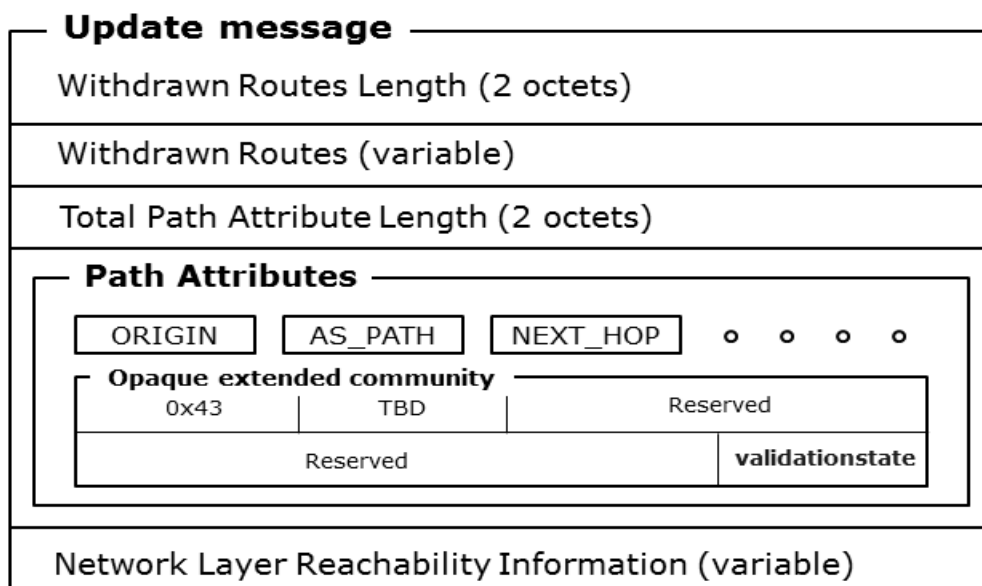
It is very important for network administrators to block bogus prefixes as soon as they detect the bogus prefixes. In case of a big company such as Google, YouTube, Amazon, and so on, a considerable number of Internet packets come and go every single second. However, the network administrators cannot keep paying attention to their routing table. Even though the administrators subscribe IP hijacking alarm service and receive an alarm message through email, it takes time for the network administrators to block the bogus prefixes by using command line interface. In order to detect the bogus prefixes and notify iBGP peers of the bogus prefixes, the opaque extended community attribute, which is including validation state, is added to the update message and the update message is forwarded to iBGP peers.

### **3.4 Implementation**

BGP routers keep updating their routing table by sharing their routing through an update message. Figure 17 shows the update message consists of ‘Withdrawn Routes Length,’ ‘Withdrawn Routes,’ ‘Total Path Attribute

Length,’ ‘Path Attributes,’ and ‘Network Layer Reachability Information.’ The path attributes includes a number of attributes such as ‘ORIGIN,’ ‘AS\_PATH,’ ‘NEXT\_HOP,’ ‘COMMUNITY,’ ‘EXTENDED COMMUNITY,’ and so on. The ‘EXTENDED COMMUNITY’ attribute will contain opaque extended community to carry the validation state of the update message between iBGP peers. The opaque extended community is a work in progress by Secure Inter-Domain Routing working group.

An update message can include a number of Path Attributes. So if a BGP router receives an update message that includes a bogus prefix, then the BGP router adds the opaque extended community to the update message and forwards the update message to iBGP peers. The iBGP peers can know whether the prefix is hijacked by checking the opaque extended community.

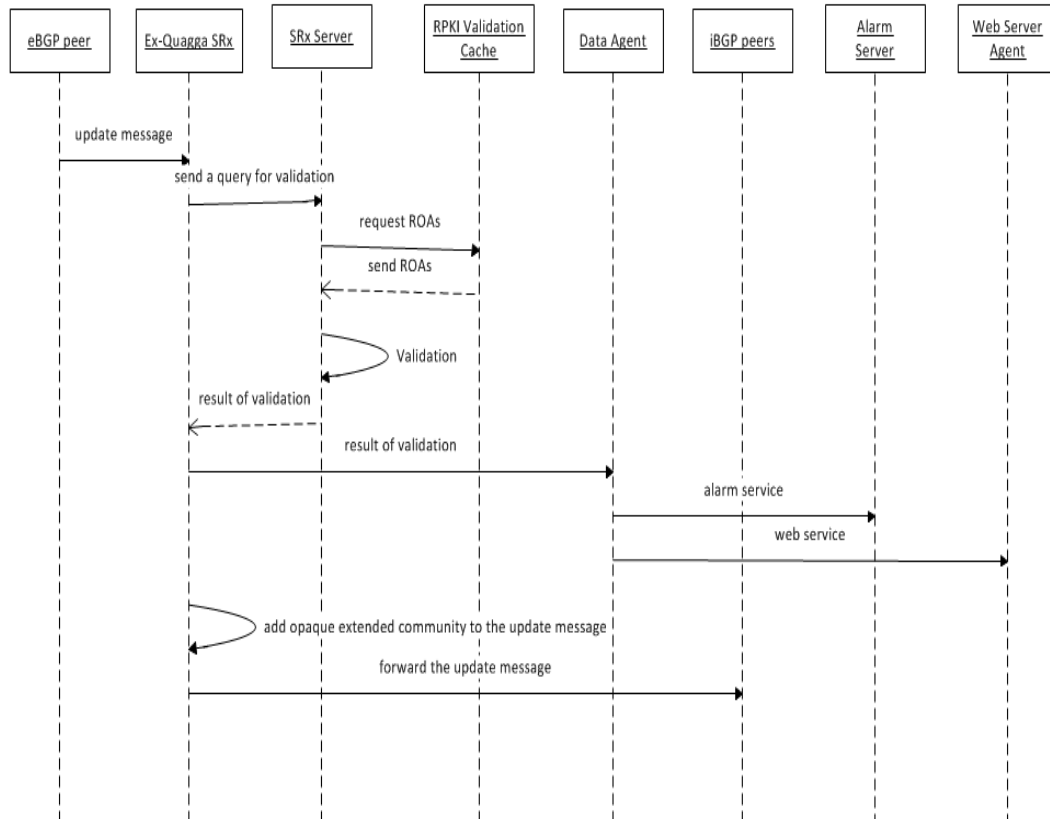


**Figure 17. The update message format**

An operation of notifying function is as follows (see Figure 18)



- The eBGP peer announces an update message
- The Ex-Quagga SRx sends a query to the SRX Server for an update validation
- The SRX server requests ROAs to the RPKI Validation Cache
- The RPKI Validation Cache provides ROAs
- The SRx Server conducts RPKI validation by comparing a prefix and an ASN to ROAs
- The SRx Server returns the result of validation to the Ex-Quagga SRx
- The Ex-Quagga SRx saves the result of validation to the Data Agent
- The Alarm Server sends notification to the Alarm System according to database in the Data Agent
- The Web Server Agent provides web interface service by using the validation information provided by the Data Agent
- The Ex-Quagga SRx adds the opaque extended community to the update message
- The Ex-Quagga SRx forwards the update message to iBGP peers
- iBGP peers update their control plane according to the update message

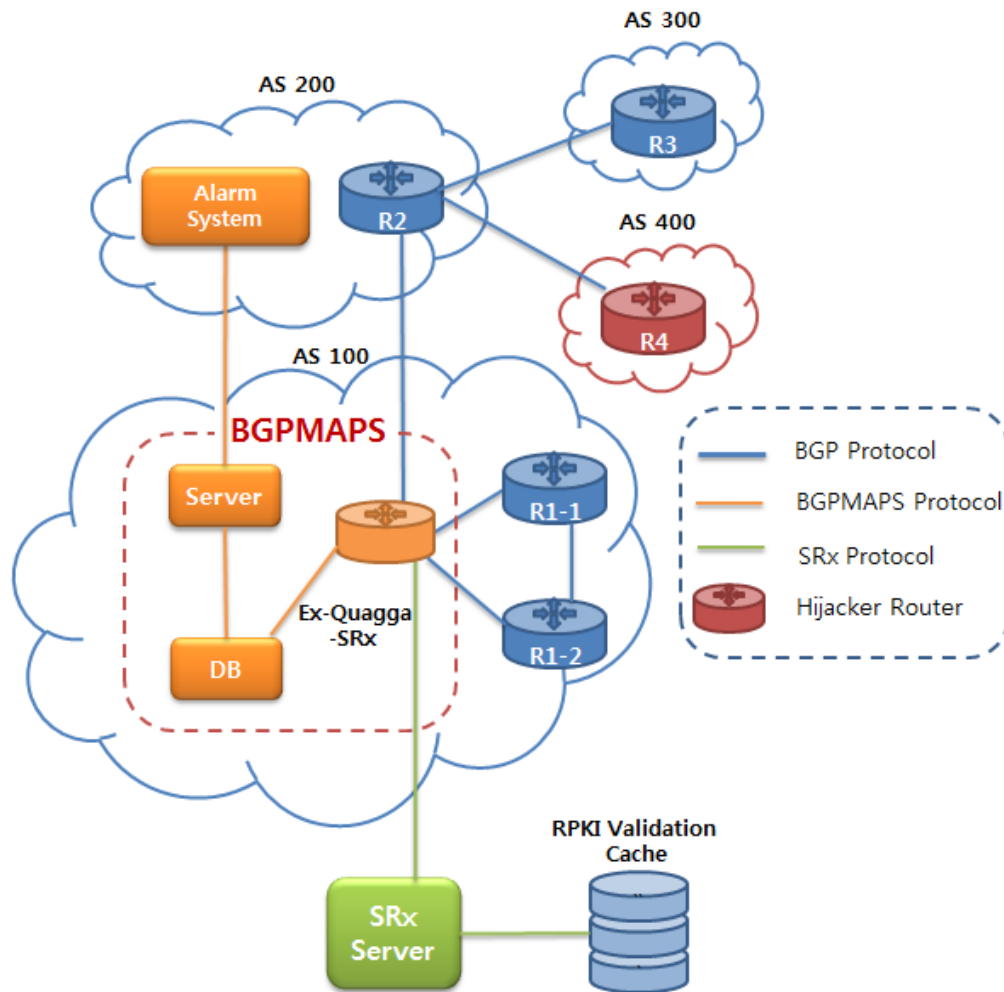


**Figure 18. A sequence diagram of the BGPMAPS**

### 3.5 Experiments

Figure 19 shows a topology that includes four ASes and each router in ASes doesn't have any capability of validating BGP update announcements. Through this topology, we explain how the AS 200 administrator can detect a bogus announcement with the Alarm System instead of installing a RPKI-enabled BGP router in AS 200. AS 300 originates 10.30.0.0/16 and the announcement is forwarded to the neighbors, AS 200, AS 400, and AS 100. Suppose AS 400 hijacks the traffic heading for AS 300 by originating 10.30.0.0/17 to AS 200. Then, R1 receives the bogus announcement and the Ex-Quagga-SRx sends a query to the SRx Server to validate the bogus

announcement.



**Figure 19. Topology for simulation**

After detecting the bogus announcement, the Ex-Quagga-SRx saves the bogus announcement in the database. The Server monitors the database and automatically makes sounds of alarm when the bogus announcement is discovered in the database. Once the sound of the alarm is made in AS 200, the AS 200 administrator can realize that there is a bogus prefix in the BGP router. In addition, the AS 200 administrator can check the bogus prefix through the

webpage that is provided by the BGPMAPS. As a result of this, the AS 200 administrator can block the bogus prefix in short time.

Figure 20 shows the BGP table in the Ex-Quagga-SRx and the BGP table was created based on the previous simulation. The value of the SRxVal indicates the result of validation after the SRx Server validates the update messages. We created an eVal column on the table to display the opaque extended community value. If the value of eVal is 0, then IP address and prefix is valid. If the value of eVal is 1, then IP address and prefix is unknown. If the value of eVal is 2, then the IP address and prefix is invalid.

	SRxVal	eVal	SRxLP	Status	Network	Next Hop	Metric	LocPrf	Weight	Path
*> u(u,-)	1				10.20.0.0/24	10.55.10.180	0		0 200	i
*> v(v,-)	0				10.30.0.0/24	10.55.10.180			0 200 300	i
*> i(i,-)	2				10.30.0.0/25	10.55.10.180			0 200 400	i
*> v(v,-)	0				10.30.1.0/24	10.55.10.180			0 200 300	i
*> v(v,-)	0				10.30.2.0/24	10.55.10.180			0 200 300	i

**Figure 20. Ex-Quagga-SRx BGP table**

Figure 21 shows the normal BGP table in case that the opaque extended community doesn't exist in the BGP update message. The normal BGP table indicates that the router will select AS 400 as the destination of 10.30.0.0/24 because the prefix 10.30.0.0/25 announced by AS 400 has a longer prefix than 10.30.0.0/24 announced by AS 300. In that case, the AS 300 is hijacked and the internet packets heading for 10.30.0.0 will be transferred to the AS 400.

Network	Next Hop	Metric	LocPrf	Weight	Path
* i10.20.0.0/24	10.55.10.180	0	100	0 200	i
* i10.30.0.0/24	10.55.10.180		100	0 200 300	i
* i10.30.0.0/25	10.55.10.180		100	0 200 400	i
* i10.30.1.0/24	10.55.10.180		100	0 200 300	i
* i10.30.2.0/24	10.55.10.180		100	0 200 300	i

**Figure 21. Normal BGP table**

Figure 22 shows R1-1 BGP table after the R1-1 receives the opaque extended community from the Ex-Quagga-SRx. Even though the R1-1 doesn't have a capability of validating update messages, the R1-1 can recognize the invalid prefix by accepting the opaque extended community. The BGP table includes the eVal to show the validation state. As a result, the R1-1 will not select AS 400 as a destination of 10.30.0.0/24.

eVal	Network	Next Hop	Metric	LocPrf	Weight	Path
* i 1	10.20.0.0/24	10.55.10.180	0	100	0 200	i
* i 0	10.30.0.0/24	10.55.10.180		100	0 200 300	i
* i 2	10.30.0.0/25	10.55.10.180		100	0 200 400	i
* i 0	10.30.1.0/24	10.55.10.180		100	0 200 300	i
* i 0	10.30.2.0/24	10.55.10.180		100	0 200 300	i

**Figure 22. R1-1 BGP table**

### 3.6 Performance test

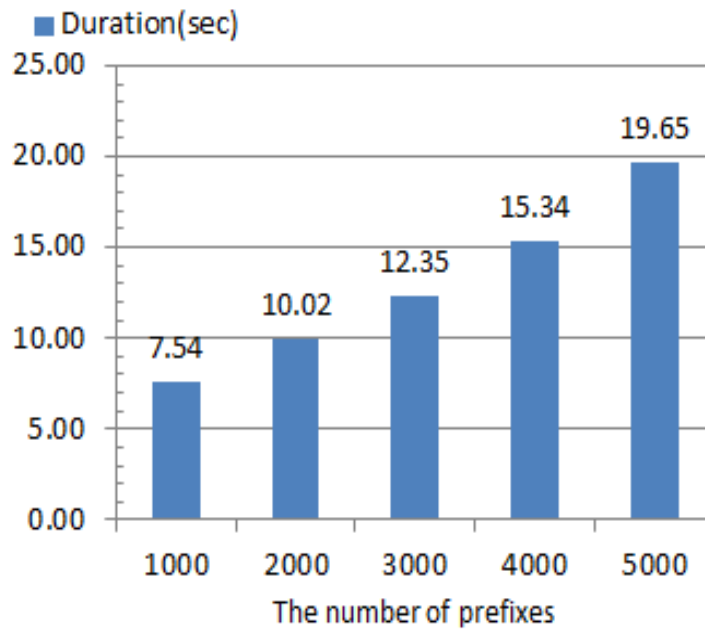
We set up a topology that is the same as Figure 19 where each router runs on a 3.40 GHz i5-3570 machine with 1 GB of memory running CentOS 6.3 and the SRx Server runs on a 3.40 GHz i5-3570 machine with 1 GB of memory running CentOS 6.3. Because it is important to quickly notify iBGP peers of bogus prefixes, we measured the duration from the moment that Ex-Quagga-SRx receives an update message to the moment that the Ex-Quagga-SRx forwards the update message, which includes the opaque extended

community, to iBGP peers. We stored 8,000 ROAs in the RPKI validation Cache because the number of ROAs in the RPKI validation Cache affects the performance and there are around 8,000 ROAs in the real world. We collected the number of prefixes that originated from each AS on the 25th of May in 2013. According to Table 3, the BGP router originates, on average, 11.74 prefixes and at most 5230 prefixes.

**Table 3. The number of prefixes in whole AS**

	Minimum	Maximum	Average
<b>Prefix(IPv4)</b>	0	4,808	11.42
<b>Prefix(IPv6)</b>	0	422	0.32
<b>Total</b>	0	5,230	11.74

We varied the number of prefixes and Figure 23 illustrates the result of the performance test. When a router originates 1,000 prefixes and the Ex-Quagga-SRx receives the prefixes, it takes 7.54 seconds. As the number of prefixes increases, the computation time increases linearly. Therefore, we can say it takes a reasonable time to notify iBGP peers of invalid prefixes.



**Figure 23. Result of the performance test**

AS administrators need to purchase RPKI-enabled BGP routers to secure their AS. However, the costs may be a burden to AS administrators to change their routers. Even if they purchase a new RPKI-enabled router or they update the existing router to have the origin validation without replacing their router, AS administrators should reconfigure all neighbors they had and prefixes they originated.

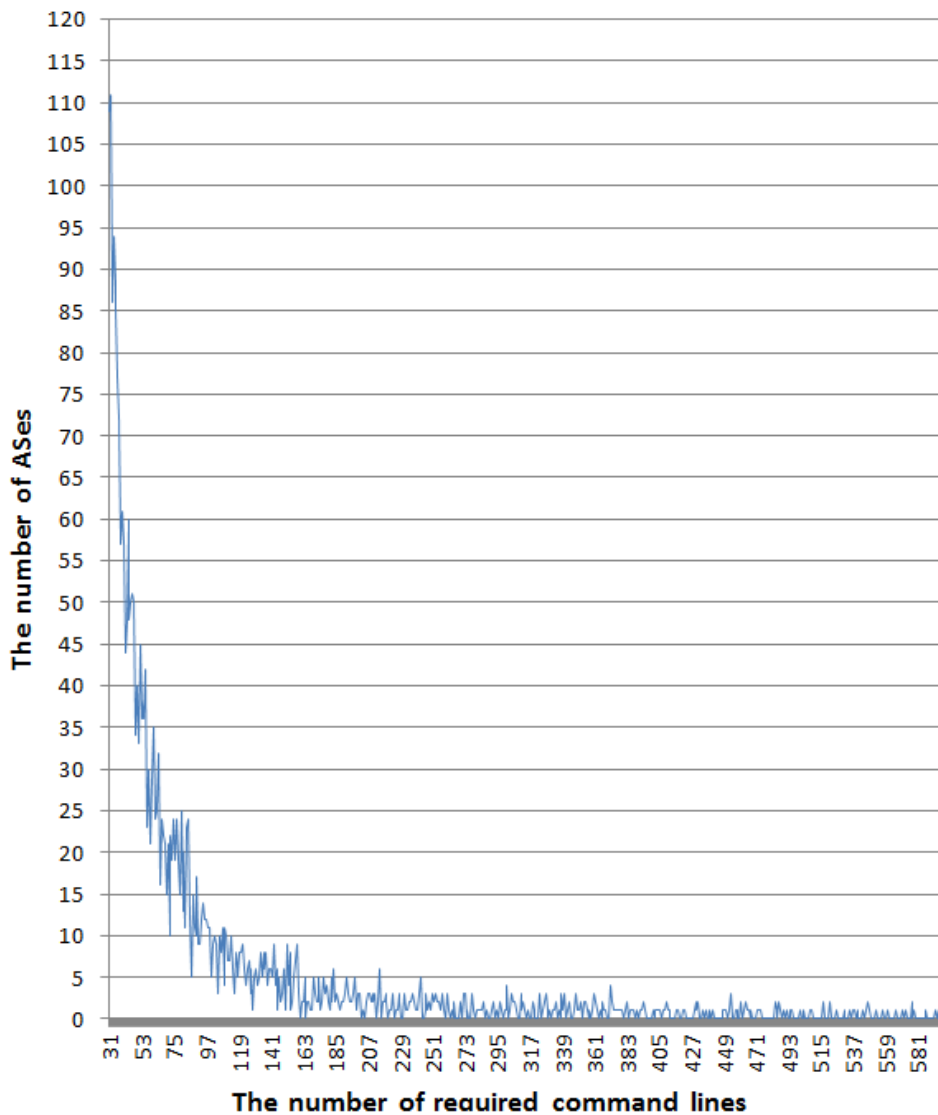
**Table 4. The number of neighbors and prefixes in whole AS**

	Minimum	Maximum	Average
Neighbor	1	3,955	5.56
Prefix(IPv4)	0	4,808	11.42
Prefix(IPv6)	0	422	0.32
Total	1	6,103	17.30

To estimate the intensity of the reconfiguration tasks, we collected the number of neighbors and the number of prefixes that originated from each AS the 25th of May in 2013, and we evaluated the average of the number of neighbors and prefixes over the whole AS. BGP data was collected by RIPE Routing Information Service (RIPE RIS) and 40,993 ASes that have at least one neighbor were used to figure out the average intensity of reconfiguration tasks. Table 4 shows the result of our analysis using collected data. The average number of neighbors for the whole AS is 5.56, which means an AS has 5 or 6 neighbors on average. The range of the number of neighbors for each AS is from 1 to 3955. However, few ASes have 200 or more neighbors. It means that most ASes have less than 200 neighbors. The average number of IPv4 prefixes for the whole AS is 11.42 and the range of the number of IPv4 prefixes for each AS is from 0 to 4,808 but few ASes have 300 or more prefixes of IPv4. The average number of IPv6 prefixes for the whole AS is 0.32 and the range of the number of IPv6 prefixes for each AS is from 0 to 422 but few ASes have 30 or more prefixes of IPv6. The total in Table 2 is a sum of neighbors and prefixes of both IPv4 and IPv6 for each AS and signifies the number of imperative reconfiguration tasks for each AS. The average of the total is 17.3, which means reconfiguration task should be done 17 times for an AS on average. The range of the number of total is from 1 to 6,103 but most ASes have less than 300 tasks. We assume more than 20 tasks are irksome and onerous enough and the number of ASes with 20 or more tasks are 5,127, that is about one-eighth of ASes we collected. Figure 24 shows the number of



required command lines for reconfiguration when the existing router is replaced with a new RPKI-enabled router.



**Figure 24. The number of required command lines of reconfiguration**

In conclusion, 5127 ASes (about one-eighth of all ASes) require more than 20 command lines to replace their BGP routers. We assume that typing 20 or more command lines is bothersome and enough to unintentionally make hijacking or mis-announcing during the reconfiguration.

## Chapter 4. The SAPBGP

### 4.1 Overview of the SAPBGP

As the number of IP prefix hijacking incidents has increased, many solutions are implemented to prevent IP prefix hijacking, such as RPKI, BGPmon, Argus, and PHAS. Except RPKI, all of the solutions proposed so far can prevent only IP prefix hijacking. The SIDR working group is researching RPKI to prevent the AS path hijacking through BGPsec, but it seems to take a long time for BGPsec to be deployed in the real world. Therefore, none of the currently available tools can prevent AS path hijacking.

In order to prevent the AS path hijacking, the proposed SAPBGP monitors the AS\_PATH attribute in update messages whether each AS in the AS\_PATH attribute is connected to each other based on our policy database that is collected from RIPE NCC repository. RIPE NCC is one of the Regional Internet Registries (RIRs) and RIPE NCC repository collects and stores Internet routing data including BGP update messages and RIB dumps. In addition, routing policies are voluntarily declared and maintained by AS holders in any of the available IRR. We conducted AS path validation using the policy-based database with the real BGP messages that are collected from the BGPmon projects. Our analysis shows 4.57% of the AS\_PATH attribute is invalid and 95.43% of the AS\_PATH attribute is valid from the fifteenth of April in 2014 to the eighth of June in 2014. In addition, the performance test

verifies that the SAPBGP can process all of the live BGP messages coming from BGPmon in real time.

## 4.2 Monitor AS Path connections

We approached differently from BGPsec to monitor and detect the AS Path hijacking by using ASes connection information using BGP peer information through policy-based database peer information. RIPE NCC provides users with RIPE Data Repository that contains BGP peer information. Through this information, we can know if any ASes are connected to other ASes. This peer information has been collected by either Routing Information Service (RIS) or Internet Routing Registry (IRR). RIS has collected and stored Internet routing data from several locations all over the world since 2001.

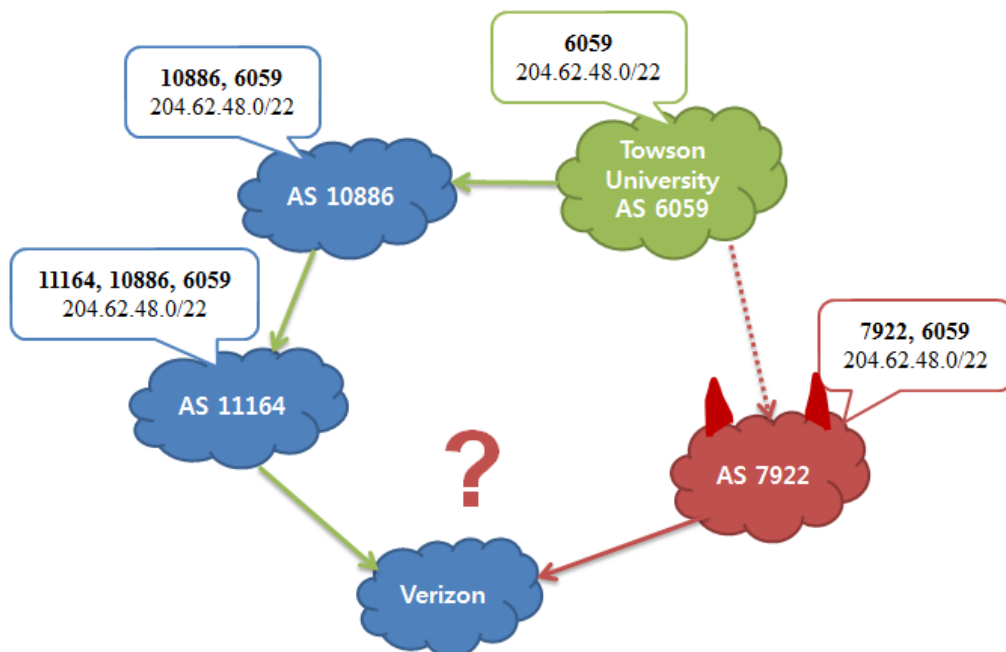


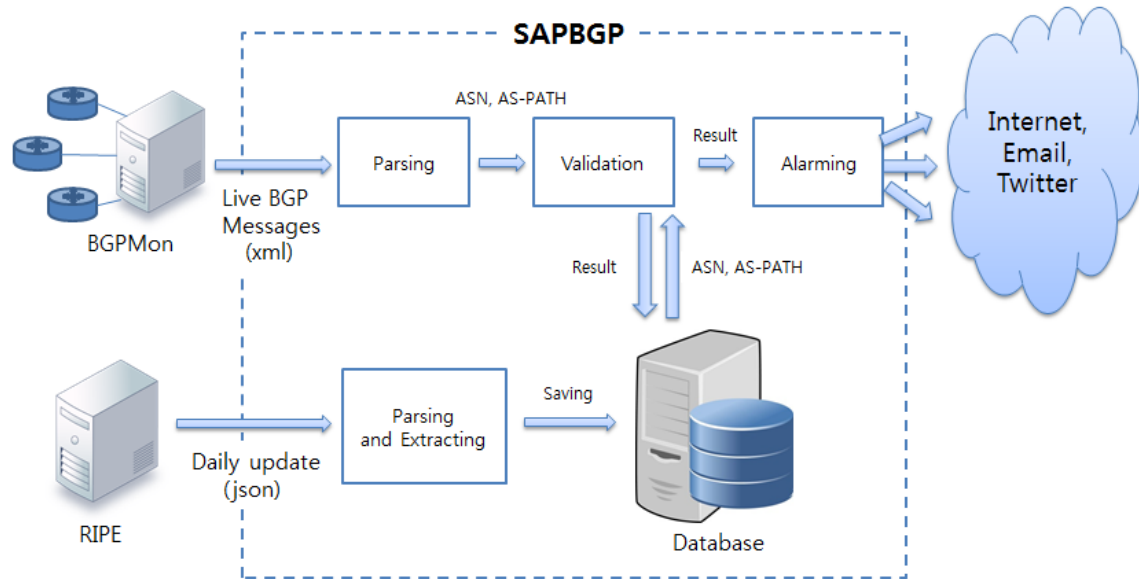
Figure 25. A scenario of manipulating a BGP message

Using peer information, the SAPBGP monitors live BGP stream from BGPmon. For example, in Figure 25, suppose that AS 7922 pretends as if AS 7922 is connected to Towson University (AS 6059), and AS 7922 creates a BGP message as if the BGP message is coming from Towson University and forwarding the BGP message. Then, Verizon cannot check that AS 7922 and Towson University are connected to each other even though Verizon can conduct the origin validation. As a result of this 1-hop hijacking, all Internet traffic that is heading for Towson University from Verizon is transferred to AS 7922 because the number of hops of passing by AS 7922 is shorter than passing by AS 11164. However, suppose that either Verizon or one of Verizon's neighbors is a BGPmon's participant and the SAPBGP can receive the live BGP stream regarding Verizon. The AS\_PATH attribute in the BGP stream will contain AS\_PATH-Verizon's AS\_PATH, including 7922 and 6059. Then, the SAPBGP can find that AS 7922 and AS 6059 are not connected to each other according to the peer information collected from RIPE NCC repository. As a result of this, Verizon's network administrator will be alerted by the SAPBGP and realize AS 7922 might be trying the 1-hop hijacking attack to draw Verizon's Internet traffic that is heading to Towson University.

### **4.3 Architecture of the SAPBGP**

We construct our own policy-based database by using API provided by RIPE NCC. We have collected, every day, all of the AS imports and exports policies information since the eighteenth of February in 2014. In addition, we

have separated tables in the database to keep the daily information as well as the accumulated routing policy information by adding new exports and imports to the existing exports and imports in the accumulate table.



**Figure 26. The architecture of the SAPBGP**

When the BGP is designed for the first time, the initial number of bits for the AS number was 16 bits, so AS number ranged from 0 to 65,535. However, the number of bits for the AS number was changed to 32 bits. After that, each RIR reserves AS numbers as indicated Table 5. We collected policy information from AS 1 to AS 394,239 and skipped unallocated AS numbers that are not indicated in Table 5.

**Table 5. 32 bits AS number allocation above 65535**

	<b>Allocation</b>	<b>The number of ASes</b>
APNIC [48]	131,072-135,580	4,509
RIPE NCC [49]	196,608-202,239	5,632
LACNIC [50]	262,144-265,628	3,485
AFRINIC [51]	327,680-328,703	1,024
ARIN [52]	393,216-394,239	1,024

We sent queries to RIPE one by one. For example, if a query is related to AS 1 then the result includes AS 1's export policies, imports policies, and prefixes in the form of json as shown in Figure 27. The routing policies that are provided by RIPE NCC are form of json where in\_bgp, in\_whois, prefix, peer, query\_starttime/query\_endtime, resource, and authority are included. The value of in\_bgp is true if the route has been seen by RIS, otherwise the value is false. The value of in\_whois is true if the route exists in IRR or whois, otherwise the value is false. The value of prefix is the list of the prefixes that have been announced by the BGP source. The value of peer is the AS number against whom AS 1 sets the routing policy.

```

{
  "status": "ok",
  "server_id": "stat-app1",
  "status_code": 200,
  "version": "1.1",
  "cached": false,
  "see_also": [],
  "time": "2014-09-23T20:41:52.047695",
  "messages": [],
  "data_call_status": "supported",
  "process_time": 432,
  "query_id": "0c0f7f2a-4362-11e4-a411-782bcb34677e",
  "data": {
    "exports": [
      {
        "in_whois": false,
        "peer": 10753,
        "in_bgp": true
      },
      {
        "in_whois": false,
        "peer": 26114,
        "in_bgp": true
      }
    ],
    "resource": "1",
    "imports": [
      {
        "in_whois": false,
        "peer": 21616,
        "in_bgp": true
      },
      {
        "in_whois": false,
        "peer": 327713,
        "in_bgp": true
      }
    ],
    "authority": "arin",
    "prefixes": [
      {
        "in_whois": true,
        "irr_sources": [
          "RADB"
        ],
        "prefix": "10.0.8.0/24",
        "in_bgp": false
      },
      {
        "in_whois": false,
        "irr_sources": "-",
        "prefix": "190.185.108.0/22",
        "in_bgp": true
      }
    ],
    "query_starttime": "2014-09-23T16:00:00",
    "query_endtime": "2014-09-23T16:00:00"
  }
}

```

**Figure 27. sample of the routing policies by the RIPE NCC API**

We used Java to implement parsing and extracting functions that parse json the form of routing policy information and extract exports and imports routing policy information. The two types of routing policy information are stored to AS 1's record in the table. As a result, a new table is created every day to keep track of the daily policy information. In addition, the accumulated table is updated by adding new policies if AS 1 adds new policies against other ASes. Figure 28 shows the records from AS 9000 to AS 9035 in the policy table.



asn	export	import
9000	34984,8685,174,12296,47123,6774,9121	34984,47123,29286
9001	5603,9119	5603,9119
9002	40964,40965,8201,24586,24588,196621,24594,32787,8218,8220,4...	40964,40965,8201,24586,24588,196621,24594,32787,8218,8220,43646,1...
9003	24963,13193,1299,16276,8218,6939,12322,9002,174,6453,3257,3...	15169,12322,24963,1301,15557,41191,13193,8426,15422,20940,5410,17...
9004	8736,16122,6875,8220	8736,16122,6875,8220
9005		
9006	41066,2854,21127	41066,2854,21127
9007	12312,20676,9211,9132	12312,9211,9132
9008	6661,44295,16265,12684,12431,197264,3347,20501,12312,42652,...	6661,44295,16265,12684,12431,197264,3347,20501,12312,42652,24611,...
9009	20485,29066,43531,29076,6830,8218,6939,286,5541,41692,9002,...	36344,57099,198722,41102,29073,43545,31128,197529,12573,33968,199...
9010		
9011	5089,1849	5089,1849
9012		
9013		
9014	2856,702,39295	2856,702
9015	5377,2614,11127	5377,2614,11127
9016	5617,34254	5617,34254
9017	3292,3308	3292,3308
9018	6805,1270	6805,1270
9019		
9020	9063,12292,1275,42652,21473,199578	12292,1275,42652,199578
9021	20864,39298,49027,29060,6663,12301,34959,51739,9121,24667,1...	20864,39298,49027,29060,6663,12301,34959,51739,9121,24667,197042,...
9022	21473,12843,3356,31334	21473,12843,3356,31334
9023	1764,8447	1764,8447
9024		
9025	6730,3303,7018	6730,3303,7018
9026	8928,5602,13284,49605,43531,6939,12637,12874,5580,20485,6730	5602,13284,49605,5580,6939,51580,12637,2686,15589,50809,15469,493...
9027		
9028	8866,8390,12615,20876,8717,9070,38932	8866,8390,12615,20876,8717,9070
9029	2917,3215	2917,3215
9030	8918,16119	8918,16119
9031	34305,44034,8708,34309,15879,31242,6667,28685,20495,16265,4...	34305,44034,8708,34309,15879,31242,6667,28685,20495,16265,44050,3...
9032	20483,9110,25478,58288,8331,15756,3216,2578,5523,29076,8470...	20483,8810,25478,8331,15756,3216,2578,5523,29076,8470,9110,12314,1...
9033	198913,57353,1547,12306,20825,29208,15388,16218,38948,3536...	198913,57353,1547,12306,20825,29208,15388,16218,38948,35366,40999...
9034	12779,16004,12551,8968,6665,8718,12909,5392,8722,5395,5396,...	12779,16004,12551,8968,6665,8718,12909,5392,8722,5395,5396,5397,83...
9035	1267,6762,1299,1239,3320,1273,3356	1267,6762,1299,1239,3320,1273,3356

**Figure 28. A screen capture of the policy table**

BGPmon provides live BGP streams through telnet to the public for research purposes, which is [livebgp.netsec.colostate.edu](http://livebgp.netsec.colostate.edu) port 50001[25]. So, whenever the routers that are connected to BGPmon receives BGP update messages, BGPmon converts BGP update messages to XML format messages and propagates the XML format messages to their clients. In addition, anyone can participate the BGPmon project to provide BGP live streams to the public. Currently, there are 9 organizations as indicated in Table 6.

**Table 6. 9 organizations who participated in the BGPmon project**

<b>AS number</b>	<b>Organization name</b>
812	Rogers Cable Communication Inc.
3303	Swisscom (Switzerland) Ltd
3257	Tinet SpA (RIPE NCC)
5568	ROSNIROS Russian Institute for Public Networks
6447	University of Oregon
10876	MAOZ.com
14041	University Corporation for Atmospheric Research
12145	Colorado State University
28289	Americana Digital Ltda.

Apart from all of the attributes that are included in the BGP update message, the XML format message from BGPmon live streams includes timestamp, date time, BGPmon id, BGPmon sequence number, and so on. Figure 29 shows the example of BGPmon live message. We used java to implement a parsing function that parses live xml stream and extracts AS number and AS\_PATH attributes.

```

<BGP_MESSAGE length="00002806" version="0.4" xmlns="urn:ietf:params:xml:ns:xb-0.4"
  type_value="2" type="UPDATE">
    <BGPMON_SEQ id="127893688" seq_num="1954792938"/>
    <TIME timestamp="1392668243" datetime="2014-02-17T20:17:23Z" precision_time="0"/>
    <PEERING as_num_len="4">
      <SRC_ADDR>
        <ADDRESS>187.16.216.68</ADDRESS>
        <AFI value="1">IPv4</AFI>
      </SRC_ADDR>
      <SRC_PORT>179</SRC_PORT>
      <SRC_AS>28347</SRC_AS>
      <DST_ADDR>
        <ADDRESS>200.160.6.217</ADDRESS>
        <AFI value="1">IPv4</AFI>
      </DST_ADDR>
      <DST_PORT>179</DST_PORT>
      <DST_AS>6447</DST_AS>
      <BGPID>0.0.0.0</BGPID>
    </PEERING>
    <ASCII_MSG length="109">
      <MARKER length="16">FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</MARKER>
      <UPDATE withdrawn_len="16" path_attr_len="66">
        <WITHDRAWN count="4">
          <PREFIX label="WITH">
            <ADDRESS>141.101.254.0/24</ADDRESS>
            <AFI value="1">IPv4</AFI>
            <SAFI value="1">UNICAST</SAFI>
          </PREFIX>
          <PREFIX label="WITH">
            <ADDRESS>141.101.253.0/24</ADDRESS>
            <AFI value="1">IPv4</AFI>
            <SAFI value="1">UNICAST</SAFI>
          </PREFIX>
          <PREFIX label="WITH">
            <ADDRESS>109.161.64.0/20</ADDRESS>
            <AFI value="1">IPv4</AFI>
            <SAFI value="1">UNICAST</SAFI>
          </PREFIX>
          <PREFIX label="WITH">
            <ADDRESS>204.245.102.0/24</ADDRESS>
            <AFI value="1">IPv4</AFI>
            <SAFI value="1">UNICAST</SAFI>
          </PREFIX>
        </WITHDRAWN>
        <PATH_ATTRIBUTES count="5">
          <ATTRIBUTE length="1">
            <FLAGS transitive="TRUE"/>
            <TYPE value="1">ORIGIN</TYPE>
            <ORIGIN value="0">IGP</ORIGIN>
          </ATTRIBUTE>
          <ATTRIBUTE length="22">
            <FLAGS transitive="TRUE"/>
            <TYPE value="2">AS_PATH</TYPE>
            <AS_PATH>
              <AS_SEG type="AS_SEQUENCE" length="5">
                <AS>28347</AS>
                <AS>262589</AS>
                <AS>2914</AS>
                <AS>701</AS>
                <AS>11486</AS>
              </AS_SEG>
            </AS_PATH>
          </ATTRIBUTE>
          <ATTRIBUTE length="4">
            <FLAGS transitive="TRUE"/>
            <TYPE value="3">NEXT_HOP</TYPE>
            <NEXT_HOP>187.16.216.68</NEXT_HOP>
          </ATTRIBUTE>
          <PATH_ATTRIBUTES>
            <NLRI count="1"><PREFIX label="DPATH"><ADDRESS>216.109.107.0/24</ADDRESS><AFI value="1">IPv4</AFI><SAFI
            value="1">UNICAST</SAFI></PREFIX></NLRI>
          </UPDATE>
        </ASCII_MSG>
      </OCTET_MSG>
      <OCTETS
        length="109">FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF006D020010188D65FE188D65FD146DA14018CCF566004240010100400
        006EBB000401BD00000B62000002BD000002CDE400304BB10D844C008100B6201A40B6203EE0B6207D00B620BB8E01008
        1BD012C18D86D6B</OCTETS>
      </OCTET_MSG>
    </BGP_MESSAGE>

```

Figure 29. BGPmon sample data

We measured the number of update messages that BGPmon propagates for one day on February in 2014. Table 7 shows the minimum, maximum, and average number of update messages per 10 seconds.

**Table 7. The number of BGP Update Messages from BGPmon Project**

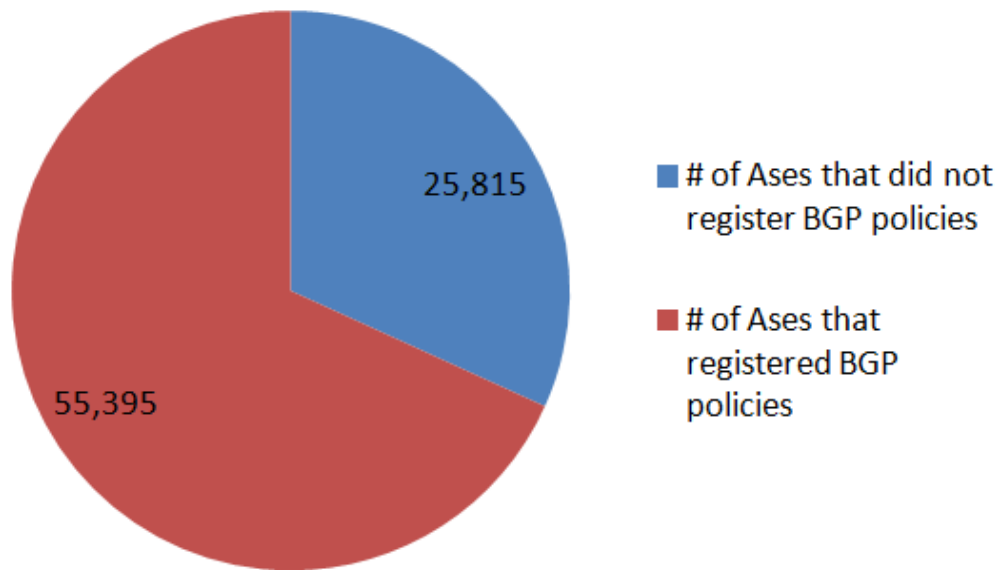
	<b>The number of update messages per 10 seconds</b>
Minimum	5
Maximum	1,321
Average	28.13

After parsing the live BGP message, the SAPBGP retrieves the ASN attribute and the AS\_PATH attribute to check whether ASes in the AS\_PATH attribute are connected to each other. Firstly, we compare the policy table in the database that is collected one day before. If a pair of ASes in the AS\_PATH is not found in the table, we compare the pair of ASes to AS policy information from the accumulated table. If we cannot find the pair from the table, we consider the AS\_PATH attribute as the suspicious AS\_PATH attribute. If we find a suspicious AS\_PATH attribute, we notify the AS network administrators of the suspicious AS\_PATH attribute.

#### **4.4 Experiments**

In order to monitor AS path hijacking in the real world, we collected BGP live stream from the BGPmon project and compared the AS\_PATH attribute to our policy-based database. The policy-based database is updated daily because BGP policy information changed whenever network operators

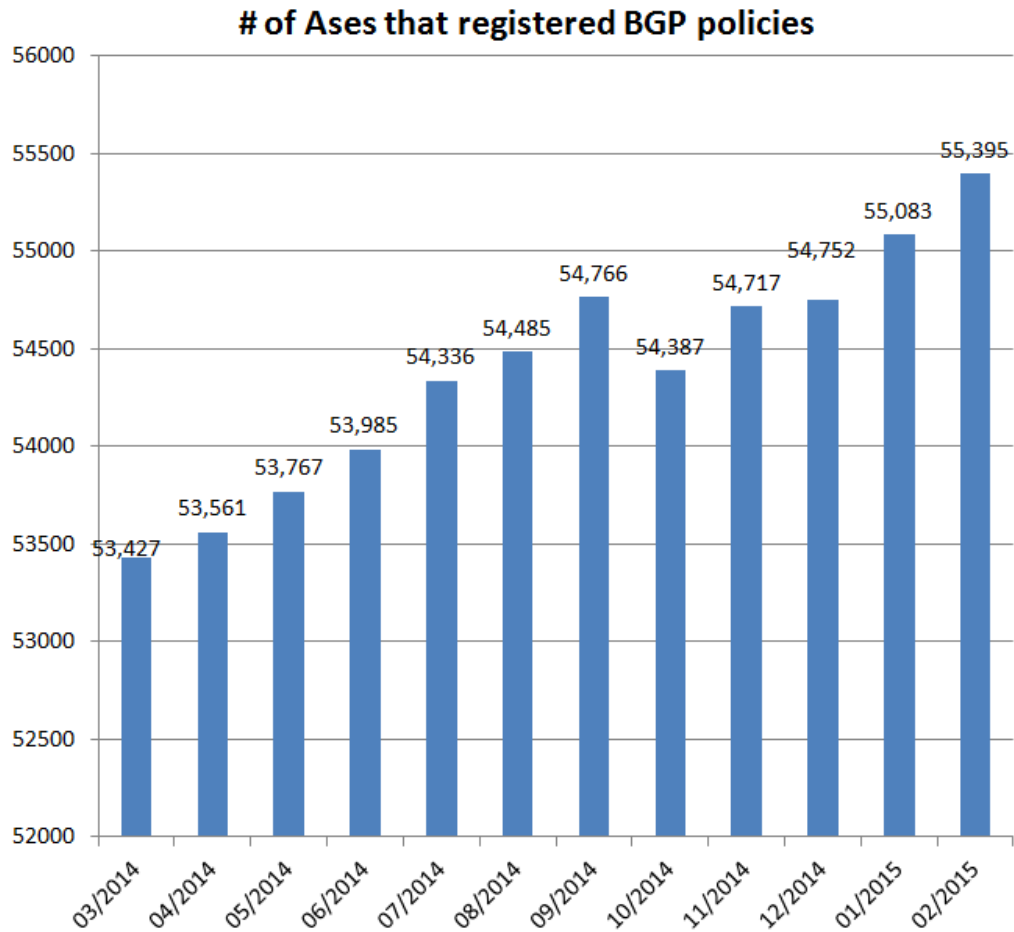
wanted to change their BGP policies. A new BGP policy table is created every day, so we used the BGP policy table that is collected one day before the day we conducted experiment. The number of BGP routing policies that are registered by AS holders is 55,395 on February in 2015, which means only 68% of AS holders registered their BGP routing policies as shown in Figure 30.



**Figure 30. Ratio of ASes that registered BGP routing policies**

The number of ASes that registered BGP routing policies are gradually increased according to our policy database. The total number of ASes is 81,210 and it will take a long time for every AS holder to register BGP policies. Figure 31 shows how many of ASes that registered BGP policies is increased for 1 year between March in 2014 and February in 2015. In order to check connection between two peers, BGP policy information from each BGP peer should contain the BGP policy against the other peer. However, we considered BGP connection is valid if only one of two BGP peers has the BGP

policy against the other peer because the number of ASes that registered BGP policy is still small. In addition, we considered a BGP message as valid message if one of an AS\_PATH pair is the one of 9 organizations that participates in the BGPmon project.

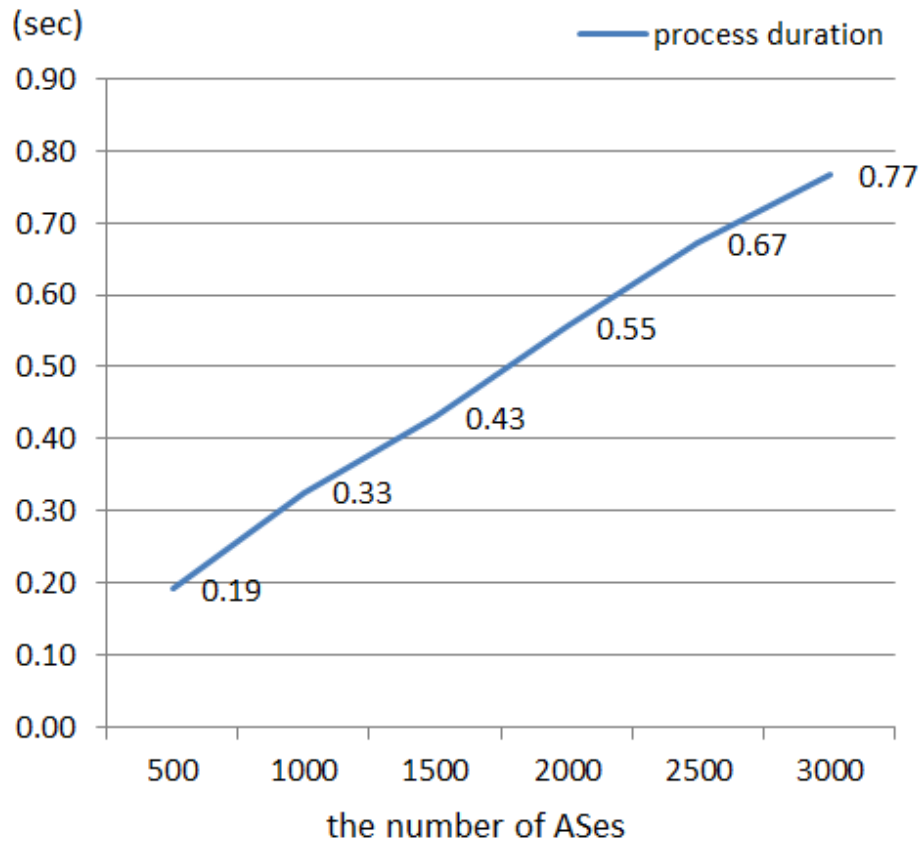


**Figure 31. # of ASes that registered BGP policies**

#### 4.5 Performance test

The SAPBGP runs on a 3.40 GHz i5-3570 machine with 16 GB of memory running Windows 7. MySQL Ver. 14.14 Distrib 5.1.41 is used for the database. We used JAVA to implement the SAPBGP that collects daily

updates from RIPE NCC, receives live BGP streams from BGPmon, and validates the BGP stream by comparing the AS\_PATH attribute to our database. The SAPBGP and database are located in the same machine to reduce the connection latency between them.



**Figure 32. The result of the performance test for the AS\_PATH validation**

Figure 32 shows the AS\_PATH validation time. The validation time includes accessing time to database, retrieving the specific AS record from a table, and comparing the AS\_PATH attribute to the AS's record. We conducted performance tests for around 1,864,567 pairs of ASes from the live BGP streams. As shown in Table 8, it takes 4.12 ms, on average, to validate a pair of ASes.

**Table 8. AS\_PATH Validation Time To Process One BGP Update Message**

	<b>Duration for verifying a BGP message</b>
Minimum	0.07ms
Maximum	9.86sec
Average	4.12ms

According to Table 7, the maximum number of live BGP messages for 10 seconds is 1,321. The SAPBGP can process 2,427.18 BGP messages for 10 seconds, on the average, based on the performance test as shown in Table 8. So, the SAPBGP can process all of the live BGP messages coming from BGPmon in real time.

#### **4.6 Result analysis**

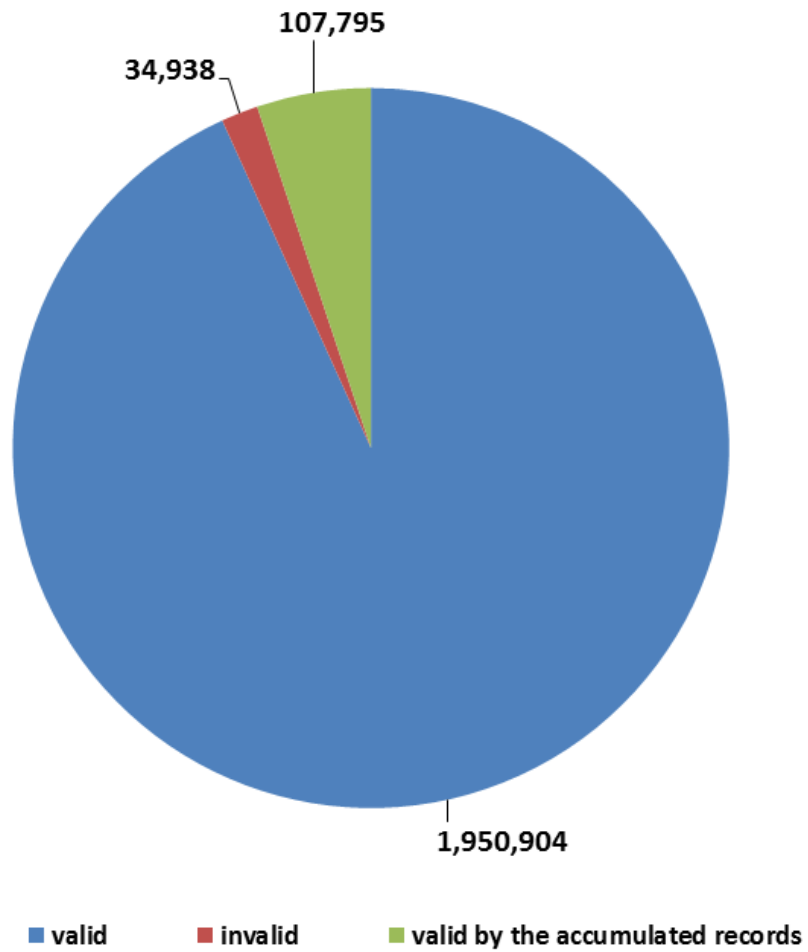
We have constructed our database by collecting daily BGP policy records from the RIPE repository since the eighteenth of February in 2014. Based on the policy-based database, the SAPBGP checked the live BGP streams whether or not pairs of AS\_PATH attributes are connected. Table 9 shows the comparison between the original results and the result that does not contain duplications from the ninth of February in 2014 to the fifth of February in 2015. Because of the difference of variation of BGP update periodic time, some pairs of ASes can be duplicated more than others.



**Table 9. The Comparison of the Results**

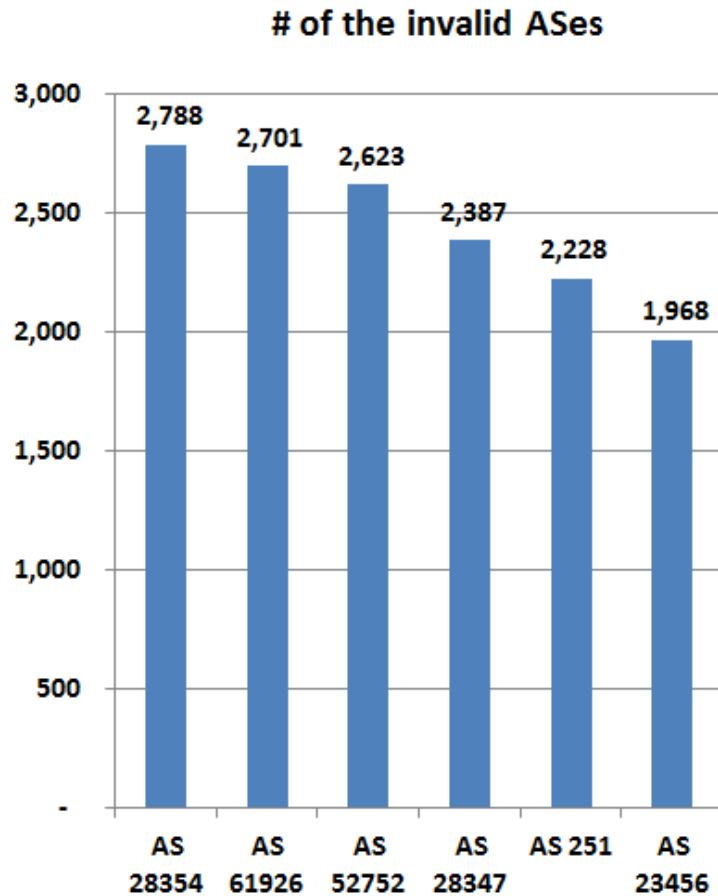
	<b>Original results</b>	<b>No duplication</b>
Valid	1,950,904	83,636
Invalid	34,938	5,271
Valid by the accumulated records	107,795	5,463

Figure 33 shows the result of the AS\_PATH monitoring experiment through the SAPBGP from the ninth of February in 2014 to the fifth of February in 2015. We conducted the experiment twice a month randomly during that period. Figure 33 shows the original data that contains many duplicated results. Our results indicate 1.67% of the AS\_PATH attributes are invalid and 98.33% of the AS\_PATH attributes are valid.



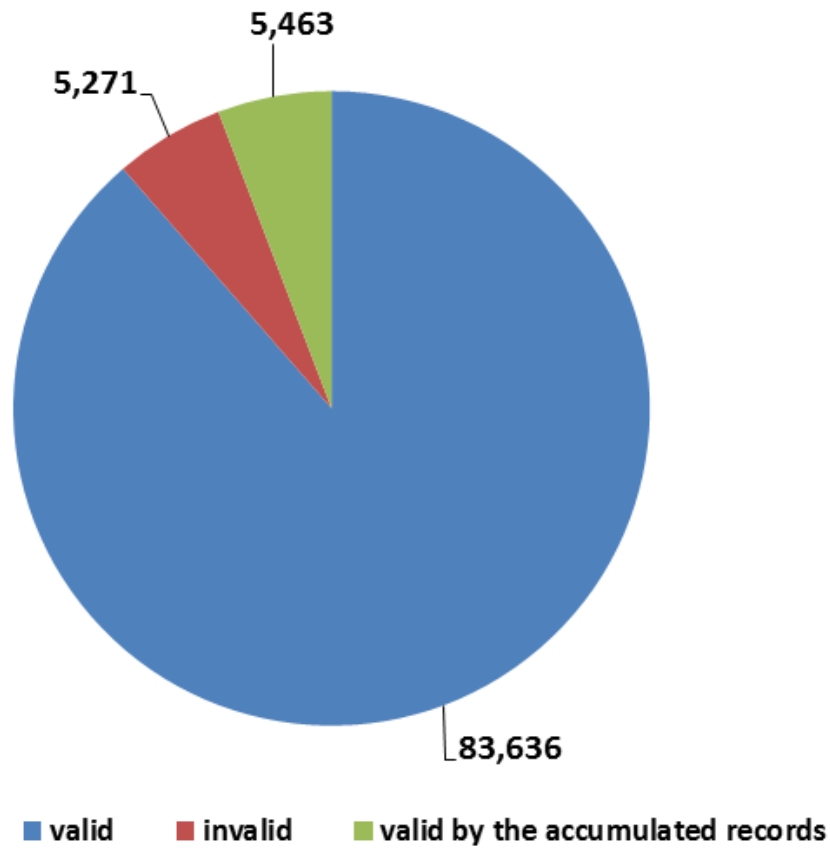
**Figure 33. The result of the AS\_PATH monitoring experiment that includes duplications.**

We conducted the experiment once a week during that period. The original data collected contains many duplicated results, but the outcome in Figure 34 does not contain the duplications. Our result shows 4.57% of the AS\_PATH attributes are invalid and 95.43% of the AS\_PATH attributes are valid. Figure 34 illustrates a portion of the policy table of the invalid ASes that the SAPBGP detected in the experiment and this result contains duplications. The invalid ASes could signify either the AS holder does not configure policies or the AS\_PATH attribute was manipulated by hijackers.



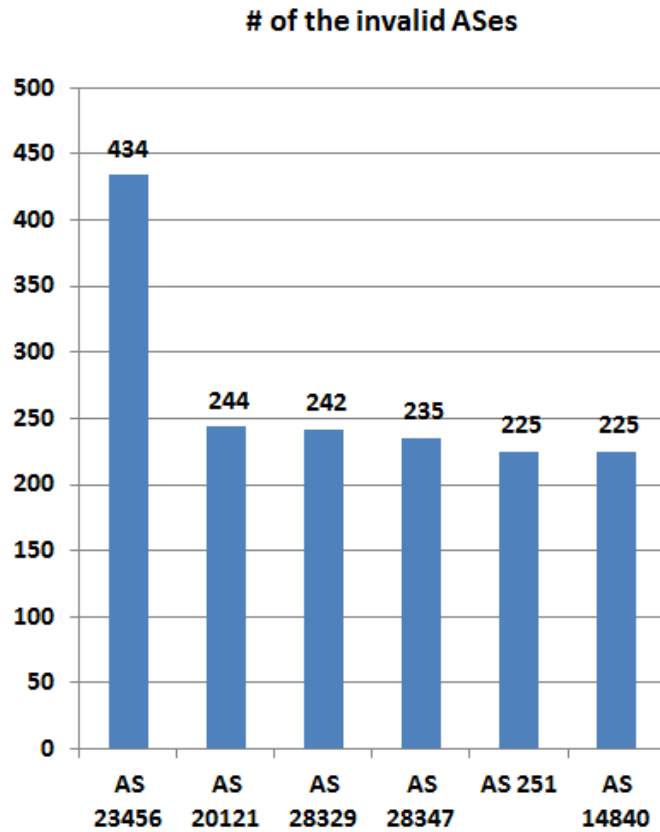
**Figure 34. A portion of the routing policy table of invalid ASes that include duplications**

Since original data contains a lot of duplicated information, we analyzed the result that does not contain duplications as well. Figure 35 shows the result of AS\_PATH that does not contain the duplications. Our result shows 5.57% of the AS\_PATH attributes are invalid and 95.43% of the AS\_PATH attributes are valid.



**Figure 35. The result of the AS\_PATH monitoring experiment that does not include duplications**

Figure 36 illustrates a portion of the policy table of the invalid ASes that the SAPBGP detected in the experiment. The result does not contain duplications from the original results.



**Figure 36. A portion of the routing policy table of invalid ASes that does not include duplications**

We analyzed the number of AS hops in AS\_PATH attribute based on our AS\_PATH validation results. Table 10 shows the number of AS hops is 5.12 on the average.

**Table 10. Analysis of the number of AS hops in AS\_PATH attributes**

	The number of ASes in the AS_PATH attribute
Minimum	2
Maximum	13
Average	5.12

We assumed that a pair of AS\_PATH that is invalid and is placed at the second position in the AS\_PATH attribute can be a candidate of 1-hop

hijacker because the number of hops should be shorter than others to draw Internet traffic to their AS. Since the first position is the destination AS, the second position AS can hijack Internet traffic heading for the first position AS. Table 11 enumerates the top 20 1-hop hijacking candidates. We checked 94,370 invalid pairs of AS\_PATH attributes that do not include duplications and we considered them as 1-hop hijacking candidates if the pair was located at first and second positions in the AS\_PATH attribute.

**Table 11. Top 20 1-hop hijacking candidates**

First position	Second position	Frequency
AS 4739	AS 3491	12
AS 4739	AS 1239	12
AS 4739	AS 1273	12
AS 4739	AS 1299	12
AS 3491	AS 7575	12
AS 4739	AS 209	12
AS 10026	AS 3491	11
AS 10026	AS 1273	11
AS 4739	AS 24115	11
AS 4739	AS 9488	11
AS 53237	AS 12956	11
AS 7575	AS 24490	11
AS 4739	AS 2914	11
AS 4826	AS 2828	11
AS 4739	AS 4635	10
AS 38809	AS 2914	10
AS 4826	AS 9498	10
AS 4739	AS 10026	10
AS 10026	AS 1299	10
AS 53237	AS 3549	10

## Chapter 5. Conclusion and Future Research

### 5.1 Conclusion

The BGPMAPS provides BGP routers with the ability to validate the prefix origin without replacing or reconfiguring their routers. AS administrators can easily have the RPKI-based origin validation function by adding the BGPMAPS to their router. The BGPMAPS continually monitors prefixes of the AS in which the BGPMAPS is installed, and notify the administrators of hijacking attempts immediately. Hence, AS administrators can filter out malicious prefixes in a timely manner. In addition to security on the BGPMAPS-enabled AS, the BGPMAPS can let other ASes know malicious prefixes through the alarm system. The BGPMAPS is economical because there is no need to purchase new RPKI-based BGP routers and it is also useful because it, compares to the new BGP router, prevents unintentional mistakes during the reconfiguration as well as spends less installation steps than reconfiguration steps of the new BGP router.

As the number of IP hijacking incidents increases, many IP hijacking monitoring tools are implemented, but network administrators still should pay attention to their routing table to protect their routers by using command line interface when the network administrator receives any warning from BGP hijacking monitoring tools because none of the monitoring tools can directly access the data plane of BGP routers. As the number of ASes and prefixes continuously increase, checking the routing information in their routers

manually is one of the big burdens on the administrators. In addition, when IP hijacking occurs, it is very important for the administrator to quickly block the bogus prefixes. Through our performance test, we showed iBGP peers can recognize the invalid prefixes in a reasonable time by accepting the opaque extended community even though the iBGP peer doesn't have a capability of validating update messages. As a result, when IP hijacking occurs, the bogus prefixes can be blocked in a timely manner, which makes the ASes more secure. We automatically prevent IP hijacking within iBGP peers, though eBGP should still be monitored in person.

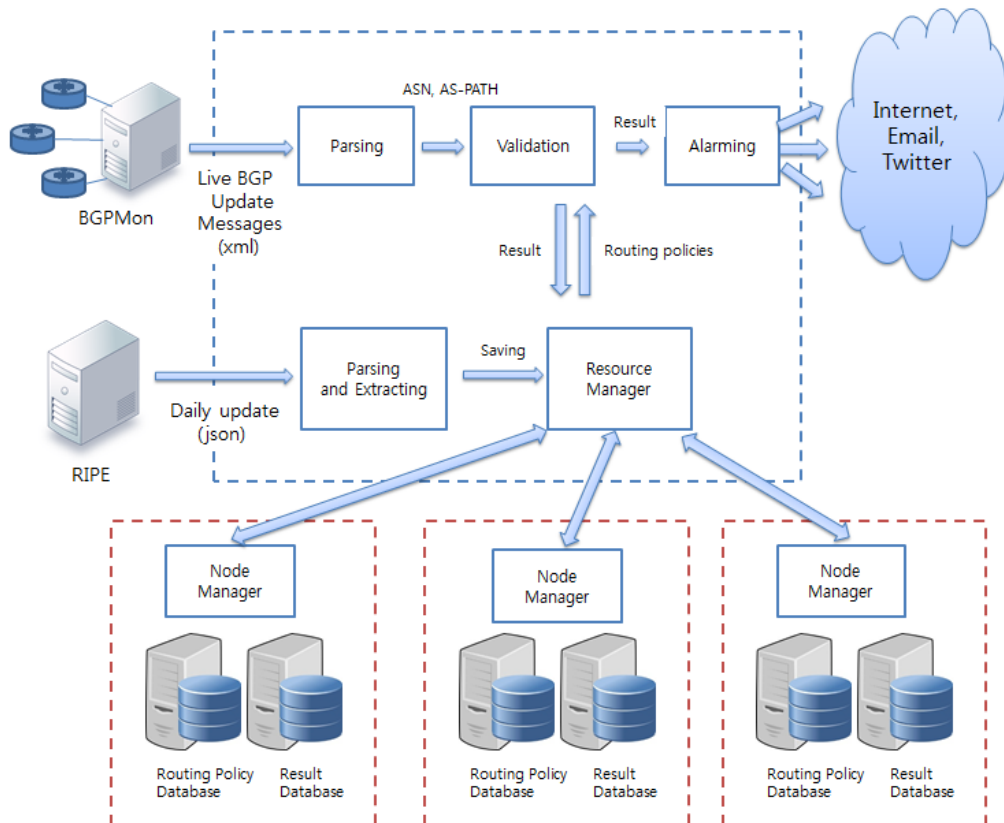
Even though many solutions are proposed to prevent IP prefix hijacking, such as RPKI, BGPmon, Argus, and PHAS, these solutions cannot protect the AS path hijacking except RPKI. SIDR proposed the RPKI using BGPSEC, but BGPSEC is currently a work in progress. In order to monitor the AS path hijacking, we propose Secure AS\_PATH BGP (SAPBGP) in which we monitor the AS\_PATH attribute in update messages whether each AS in the AS\_PATH attribute is connected to each other based on our policy database collected from RIPE NCC repository. The result of the AS\_PATH validation test shows 4.57% of the AS\_PATH attribute is invalid and 95.43% of the AS\_PATH attribute is valid from the fifteenth of April in 2014 to the eighth of June in 2014. In addition, the result of the performance test verifies that the SAPBGP can process all of the live BGP messages coming from BGPmon in real time. In the result of the AS\_PATH monitoring experiment, the ratio of invalid AS\_PATH attributes is high because some AS routers still do not



configure their policies. For the precise result of the policy based AS\_PATH validation, every router needs to configure policies against its peers.

## 5.2 Future Research

As the number of devices that require an IP address increases, the number of ASes increases. As a result, the SAPBGP should be able to handle a huge amount of BGP routing policy information. In the current system, the BGP routing policy information is stored in the local database. In order to maintain a huge amount of BGP routing policy information, the BGP routing policy information should be stored in multiple distributed databases as shown in Figure 37.



**Figure 37 distributed database for BGP routing policy information**

The Resource Manager(RM) communicates with the Node Managers(NMes) to store BGP routing policy information in two of the NMes. Even though BGP policy routing information is removed, the RM can collect the BGP policy routing information from the other NM.

Furthermore, it is important to reduce calculation time to compare pairs of ASes in the AS\_PATH attribute to BGP policy information from our policy database. Currently, the SAPBGP is able to produce the validation result in real time. However, as the number of ASes gradually grows, algorithms for comparing pairs of ASes in the AS\_PATH attribute to BGP policy information from our policy database should be improved in the future.

## REFERENCES

- [1] IANA. Internet Assigned Numbers Authority. [Online]. Available:  
<http://www.iana.org>
- [2] K. Lougheed and Y. Rekhter, "Border Gateway Protocol (BGP)." RFC 1105 (Experimental), June 1989. Made obsolete by RFC 1163.
- [3] D. Mills, "Exterior Gateway Protocol formal specification." RFC 904 (Historic), Apr. 1984.
- [4] K. Lougheed and Y. Rekhter, "Border Gateway Protocol (BGP)." RFC 1163 (Historic), June 1990. Made obsolete by RFC 1267.
- [5] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)." RFC 1654 (Proposed Standard), July 1994. Made obsolete by RFC 1771.
- [6] Y. Rekhter and T. Li. "A border gateway protocol 4 (BGP-4)". RFC 1771, Internet Engineering Task Force, Mar. 1995.
- [7] Rekhter, Y. 2006. A Border Gateway Protocol 4 (BGP-4). RFC 4271.
- [8] Murphy, S. 2006. BGP Security Vulnerabilities Analysis. RFC 4272.
- [9] "7007 Explanation and Apology," Apr 1997,  
<http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html>.

- [10] Renesys Blog, Con-Ed Steals the 'Net. [Online]. Available:  
[http://www.renesys.com/blog/2006/01/coned\\_steals\\_the\\_net.shtml](http://www.renesys.com/blog/2006/01/coned_steals_the_net.shtml)
- [11] Renesys Blog, Internet-Wide Catastrophe Last Year[Online].  
 Available:  
[http://www.renesys.com/blog/2005/12/internetwide\\_nearcatastrophela.shtml](http://www.renesys.com/blog/2005/12/internetwide_nearcatastrophela.shtml)
- [12] Renesys Blog, Pakistan hijacks YouTube [Online]. Available:  
[http://www.renesys.com/blog/2008/02/pakistan\\_hijacks\\_youtube\\_1.shtml](http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml)
- [13] P. Litke and J. Steward, "BGP Hijacking for Cryptocurrency Profit" [Online]. Available: <http://www.secureworks.com/cyber-threat-intelligence/threats/bgp-hijacking-for-cryptocurrency-profit> [Accessed February 2015]
- [14] China's 18-Minute Mystery [Online]. Available:  
<http://research.dyn.com/2010/11/chinas-18-minute-mystery/>
- [15] Manderson, T., Vegoda, L., and Kent, S. 2012. Resource Public Key Infrastructure (RPKI) Objects Issued by IANA(Feb. 2012).  
 [Online]. Available: <http://www.rfc-editor.org/rfc/rfc6491.txt>
- [16] D. Matthews, Y. Chen, H. Yan, and D. Massey. BGP Monitoring System. Available from: <http://bgpmon.netsec.colostate.edu/>.

- [17] Xingang Shi, Yang Xiang, Zhiliang Wang, Xia Yin, Jianping Wu.  
Detecting Prefix Hijackings in the Internet with Argus. In Proc.  
of ACM IMC 2012.
- [18] Lad, M., Massey, D., Pei, D., Wu, Y., Zhang, B., and Zhang, L.  
2006. PHAS: A prefix hijack alert system. In Proceedings of the  
15th conference on USENIX Security Symposium - Volume 15  
(USENIX-SS'06), Vol. 15.
- [19] Renesys Blog, Targeted Internet Traffic Misdirection [Online].  
Available: <http://www.renesys.com/2013/11/mitm-internet-hijacking> [Accessed January 2014]
- [20] M. Lepinski, Ed., and BBN, “BGPSEC Protocol Specification,”  
Available: <http://tools.ietf.org/html/draft-ietf-sidr-bgpsec-protocol-08>.
- [21] IETF. “Secure Inter-Domain Routing (SIDR)”. Online, Sep.  
2010. Available from <http://datatracker.ietf.org/wg/sidr/>
- [22] Q. Li, Y. Hu, and X. Zhang, “Even Rockets Cannot Make Pigs  
Fly Sustainably: Can BGP be Secured with BGPsec?,” 2014.
- [23] R. Lychev, S. Goldberg, and M. Schapira, “BGP Security in  
Partial Deployment”, 2013.
- [24] The BGPmon project, <http://bgpmon.netsec.colostate.edu>,  
[Accessed 6th July 2013].
- [25] BGPmon live stream, <http://bgpmon.netsec.colostate.edu/join-the-peering.html> [Accessed 27th February 2015]

- [26] J. Moy, OSPF Version 2, RFC 2328, 1998.
- [27] D. Oran, OSI IS-IS Intra-domain Routing Protocol, RFC 1142, 1990.
- [28] G. Malkin, Routing Information Protocol, RFC 1058, 1988.
- [29] G. Malkin, RIP Version 2 Carrying Additional Information, RFC 1388, 1993.
- [30] J. Schlamp, G. Carle, and E. W. Biersack. How to prevent AS hijacking attacks. In Proceedings of the 2012 ACM Conference on CoNEXT Student Workshop (CoNEXT Student 2012), Nice, France, December 2012.
- [31] R. Housley, W. Ford, W. Polk, and D. Solo, Internet X.509 Public Key Infrastructure Certificate and CRL Profile, RFC2459, 1999.
- [32] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, Internet X.509 Public Key Infrastructure Certificate, and Certificate Revocation List (CRL) Profile, RFC5280, 2008
- [33] M. Lepinski, S. Kent, and D. Kong, "A Profile for Route Origin Authorizations (ROAs)," [Online]. Available: <http://tools.ietf.org/html/rfc6482>, [Accessed December 2012].
- [34] R. Housley, "Cryptographic Message Syntax (CMS)", [Online]. Available: <http://www.ietf.org/rfc/rfc3852.txt>, [Accessed September 2014].

- [35] R. Housley. 2004. Vigil Security Cryptographic Message Syntax  
Available: [www.ietf.org/rfc/rfc3852.txt](http://www.ietf.org/rfc/rfc3852.txt) [Accessed Dec 2014].
- [36] NIST Blog, Global RPKI Repository Analysis, [Online].  
Available: <http://rpki-monitor.antd.nist.gov/?p=0&s=1> [Accessed  
February 2015].
- [37] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol  
(S-BGP)," IEEE J. Sel. Areas Commun., vol. 18, no. 4, Apr.  
2000.
- [38] R. White, "Securing BGP through secure origin BGP," Internet  
Protocol Journal, vol. 6, no. 3, September 2003.
- [39] P. van Oorschot, T. Wan and E. Kranakis, "On Interdomain  
Routing Security and Pretty Secure BGP (psBGP)," ACM  
Transactions on Information and System Security, vol. 10, no. 3,  
July 2007.
- [40] 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  
Proceedings of Internet Society Symposium on Network and  
Distributed System Security (NDSS 03), February 2003.
- [41] Josh Karlin, Stephanie Forrest, and Jennifer Rexford: Pretty  
Good BGP: Improving BGP by Cautiously Adopting Routes In  
IEEE International Conference on Network Protocols (2006)

- [42] Y.-C. Hu, A. Perrig, and M. Sirbu, "SPV: secure path vector routing for securing BGP," in SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications. New York, NY, USA: ACM, 2004, pp. 179–192.
- [43] BGP Secure Routing Extension (BGP-SRx) by NIST [Online]. Available: <http://www-x.antd.nist.gov/bgpsrx/>
- [44] L. Blunk, "Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format," RFC 6396 , 2011.
- [45] Y.-J. Chi, R. Oliveira, and L. Zhang. Cyclops: The AS-level connectivity observatory. ACM SIGCOMM Computer Communication Review, pages 7–16, 2008.
- [46] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra, "Routing Policy Specification Language (RPSL)," RFC 2622, June 1999.
- [47] P. Mohapatra, K. Patel, J. Scudder, D. Ward, and R. Bush, BGP Prefix Origin Validation State Extended Community. draft-ietf-sidr-origin-validation-signaling-04 [Online]. Available: <https://tools.ietf.org/html/draft-ietf-sidr-origin-validation-signaling-04> [Accessed August 2014].
- [48] Asia Pacific Network Information Centre. [Online]. Available: <http://www.apnic.net/>



[49] Réseaux IP Européens (RIPE) Network Coordination Centre.

[Online]. Available: <http://www.ripe.net/>

[50] Latin American and Caribbean Network Information Centre.

[Online]. Available: <http://www.lacnic.net/>

[51] African Network Information Centre. [Online]. Available:

<http://www.afrinic.net/>

[52] American Registry for Internet Numbers. [Online]. Available:

<http://www.arin.net/>

## CURRICULUM VITAE

### Personal Information

Name: Je-Kuk Yun



### Educational Background

- |                          |  |
|--------------------------|--|
| <b>2010.02 –</b>         | <b>Towson University, Maryland, USA</b><br>Doctor of Science in Computer Science   |
| <b>2008.02 – 2010.02</b> | <b>Graduated from Towson University, Maryland, USA</b><br>Master of Science in Computer Science                            |
| <b>2007.03 – 2007.06</b> | <b>Credit Bank System, National Institute for Lifelong Education, Korea</b><br>Bachelor of Engineering in Computer Science |
| <b>2004.03 – 2007.02</b> | <b>Shinheung College, Korea</b><br>Associate Degree in Computer Science  |

### Professional publications:

- **Journal Publication**

- [1] J. Yun, B. Hong, Y. Kim, "The Policy-Based AS\_PATH Verification to Prevent 1-Hop AS Path Hijacking by Monitoring BGP Live Streams", International Journal on Advances in Security, Vol.8, 2015.

- **Conference Proceeding**

- [1] J. Yun, B. Hong, Y. Kim, "The Policy-Based AS\_PATH Verification to Monitor AS Path Hijacking", The Eighth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2014), Lisbon, Portugal, 16-20, November, 2014.
- [2] J. Yun, B. Hong, Y. Kim, "The Implementation of BGP Monitoring, Alarming, and Protecting System by a BGP-UPDATE-Based Method using ECOMMUNITY in Real Time", THE 2014 INTERNATIONAL CONFERENCE ON SECURITY & MANAGEMENT (SAM 2014), Las Vegas Nevada, USA, 21-24, July 2014.
- [3] J. Yun, B. Hong, Y. Kim, "The BGP Monitoring and Alarming System to Detect and Prevent Anomaly IP Prefix Advertisement", Research in Applied Computation Symposium (RACS 2013), Montreal, QC, Canada, 1-4 October 2013.
- [4] J. Yun, C. Byun, Y. Kim, "Architecture of the Remote Routing Validation Tool for BGP Anomaly Detection", Research in Applied Computation Symposium (RACS 2012), San Antonio, TX, U.S.A. October 23-26, 2012
- [5] K. Park, C. Byun, J. Yun, J. Chang, Y. Kim, "Context-Aware Inference (CAI) Model on Smart Computing Environment", 2012 International Conference on Information Science and Applications, Kyonggi University, Suwon, Republic of Korea, 23-25 May 2012.
- [6] J. Yun, K. Park, C. Byun, "Mobile Real-time Tracking System based on the XCREAM (XLogic Collaborative FRID/USN-Enabled Adaptive Middleware)", Software Engineering Research, Management and Applications 2011, Baltimore, Maryland, U.S.A., August 10-12, 2011
- [7] C. Byun, K. Park, J. Yun, and Y. Kim, "Design and Implementation of the Context-Aware Collaboration Framework with the XCREAM," Proc. Intl. Conf. on Smart IT Applications (SITA 2011), Seoul, Korea, 2011.
- [8] K. Park, J. Yun, C. Byun, Y. Kim, and J. Chang, "The XCREAM Framework and Collaboration Validity Tests," Proc. 1st ACIS/JNU Int. Conference on Computers, Networks, Systems, and Industrial Engineering (CNSI 2011), Jeju, Korea, 2011.

- [9] K. Park, J. Yun, Y. Kim, and J. Chang, "Design and Implementation of Scenario-based Collaborative Framework: XCREAM," Proc. Intl. Conf. on Information Science and Applications (ICISA 2010), Seoul, Korea, 2010.
- [10] K. Park, J. Yun, Y. Kim, and J. Chang, "Design and Implementation of Scenario-based Collaborative Framework: XCREAM," Proc. of Int'l Conference on Information Science and Applications (ICISA 2010), Seoul, Korea, April 2010.

## **Skills**

### **Programming Languages:**

Java, Matlab, C/C++, .Net C#, ASP .NET, ASP, PHP, JSP, HTML, Javascript, CSS, XML, JSON, AJAX, JQuery, SQL

### **Operating Systems:**

Windows, Linux, Mac

### **Databases:**

MySQL, Oracle, MSSQL

### **Servers/Services:**

Apache Tomcat Webserver, Internet Information Services (IIS), Glassfish Webserver

### **Software:**

Matlab, Netbeans, Eclipse, MySQL Workbench, Photoshop, Flash

## **Project Experience**

**2012.01 ~ 2015.02**

**Project Title: BGP Routing Information Verification Tool for Preventing Inter-Domain Routing Misbehavior in Real Time**  
**Responsibility: Project Team Leader**  
**Main Techniques Used: C, Java, jQuery, MySQL, spring framework,**

### **Details:**

- Design architecture of Remote Routing Validation Tool for BGP anomaly Detection
- Design and implementation of the BGP monitoring, alarming, and protecting system by a BGP-UPDATE-Based method using ECOMMUNITY in real time

- Design and implementation of policy-based AS\_PATH verification to monitor AS Path Hijacking

**2014.07 ~ 2015.04**

**Project Title: Automatic Crack Categorization**

**Responsibility: Full-time Programmer**

**Main Techniques Used: C#, Matlab 2014**

**Details:**

- Develop solutions for automatic identification and classification of cracks presented in pavement images captured by Adhera data collection vehicles
- Identify man-made objects, co-existing with cracks, represented as white lines and classifying four types of cracks: Transverse Crack (TC), Longitudinal Crack (LC), Block Crack (BC) and Alligator Crack (AC).
- Develop math-based highly tuned image processing techniques, graph algorithms, statistical analysis of various features, and machine learning techniques tailored for image classifications.

**2013.10 ~ 2014.08**

**Project Title: MyHealthAvenue**

**Responsibility: Full-time Programmer**

**Main Techniques Used: Java, Spring Framework, JQuery, HTML 5, CSS 3**

**Details:**

- Provide service to import health records into smartphone through email / camera
- Provide service to find users' health records anytime and anywhere

**2011.10 ~ 2013.07**

**Project Title: MySmartLSAT.com**

**Responsibility: Full-time Programmer**

**Main Techniques Used: Spring Framework, JQuery, HTML 5, CSS 3**

**Details:**

- Design and Implement web-based smart E-Learning System for LSAT Logic Game Section
- Define algorithm to support question generation process
- Define algorithm for dynamic question distribution by student's learning level

**2012.09 ~ 2013.06**

**Project Title: KSEA Server Maintenance**

**Responsibility: Full-time Programmer**

**Main Techniques Used: PHP, ASP .NET, .NET C#, MSSQL, MySQL**

**Details:**

- Server Maintenance
- Develop web-based event management system
- KSEA Election System Maintenance
- General IT Support

**2009.01 ~ 2012.01****Project Title: Mobile Real-time Tracking System based on the XCREAM (XLogic Collaborative RFID/USN-Enabled Adaptive Middleware)****Responsibility: Researcher****Main Techniques Used: RFID, JAVA, MySQL****Details:**

- Implementation of real-time automatic package tracking system to notify users of the current location of their packages through smartphone using RFID and GPS
- Implementation of athlete management System (AMS) that manages four independent applications via XCREAM
- XCREAM performance evaluation in heterogeneous environments

**2011.01 ~ 2011.09****Students****Project Title: Math Workbook for Middle School****Responsibility: Full-time Programmer****Main Techniques Used: Object-C, HTML, CSS****Details:**

- Develop a new math keyboard to get input from student through a smart device, e.g. iPad
- Two versions of the math keyboards:
- Web-based math keyboard
- Math keyboard running on iOS
- Develop an iPad e-Book app for middle school students practice math

**2011.01 ~ 2011.04****Project Title: KSEA Server Migration****Responsibility: Full-time Programmer****Main Techniques Used: PHP, ASP .NET, .NET C#, MSSQL, MySQL****Details:**

- Database Redesign and Migration: Integrate Membership Database with event database
- Configure New Server and Services
- Data backup

**2007.01 ~ 2008.10****Project Title: Web site development****Responsibility: Part-time Programmer (Freelancer)****Main Techniques Used: PHP, MySQL, Apache Server, Flash, Photoshop, jQuery, Javascript****Details:**

- Implement 3 websites: shopping mall website, department of college website, Korean Resource Center website
- Redesign and implement existing online employ management system to adopt asynchronous communication using jQuery

## Awards

- **Graduate Student Scholarship** Award (2010-2015) –Towson University, Maryland
- **Best Paper Award** (2014)- The Policy-Based AS\_PATH Verification to Monitor AS Path Hijacking by Je-Kuk Yun, Beomseok Hong, Yanggon Kim Presented during SECURWARE 2014, The Eighth International Conference on Emerging Security Information, Systems and Technologies, held in Lisbon, Portugal - November 16-20, 2014

