Secure and Scalable Deployment of Resource Public Key Infrastructure (RPKI)

Zhiwei Yan[1], Guanggang Geng[1]*, Hidenori Nakazato[2], and Yong-Jin Park[3]
[1]China Internet Network Information Center, Beijing, 100190, P. R. China
{yan, gengguanggang}@cnnic.cn
[2]Waseda University, Tokyo, 169-8555, Japan
nakazato@waseda.jp
[3]University of Malaysia Sabah, Sabah, 88400, Malaysia
yjpark@ums.edu.my

Abstract

The Border Gateway Protocol (BGP) is considered to be vulnerable to some typical security risks due to its lack of schemes to verify the received BGP messages. To address BGP security issues, Internet Engineering Task Force (IETF) proposed RPKI to verify the route origination contained in the BGP message. Currently, the standardization of basic RPKI protocol have been finished. Some organizations have deployed RPKI services and some are under the process for that. However, RPKI faces additional threats during the actual deployment especially the malfunctioning of the Certification Authority (CA) when it issues certificates bound to the resources. We analyze the threats to RPKI from the perspective of its large-scale deployment and then focus on the CA operation with empirical tests. We propose a comprehensive CA-Safeguard scheme in order to support the secure and scalable deployment of RPKI in the near future[1].

Keywords: BGP, RPKI, BGPsec, Route origination, CA-Safeguard

1 INTRODUCTION

In the current Internet, the Border Gateway Protocol (BGP) [28] is widely used for exchanging of route information between Autonomous Systems (AS). However, BGP routers silently trust all the received BGP messages from the peer nodes and makes no verification about the route origination. In reality, the route information can be easily tampered or mis-configured. This always happen in today's Internet and known as BGP hijacking or prefix hijacking[34][33]. Over the recent years, some typical BGP hijacking incidents include: in February 2008 when YouTube traffic was blocked by Pakistan Internet Service Provider (ISP), in April 2010 when more than 37,000 IP prefixes were hijacked by China Telecom, in February 2014 when Canadian ISP redirected Internet traffic, and in November 2015 when some prefixes were hijacked by Bharti Airtel. Prefix hijacking is a serious security threat on Internet and it may cause block-hole routes, traffic interception and Denial-of-Service (DoS) attacks in the Internet[7].

To efficiently handle prefix hijacking, Resource Public Key Infrastructure (RPKI) was proposed by community. The core of the RPKI architecture constitutes the trustful hierarchy of IP address and AS number allocations; and the distributed databases to store and disseminate the digital objects. Based on the standardization of RPKI in Internet Engineering Task Force (IETF), the five Regional Internet Registries (RIRs) and many organizations have started the deployment of RPKI locally or globally. However,

Journal of Internet Services and Information Security (JISIS), volume: 8, number: 1 (February 2018), pp. 31-45
*Corresponding author: Tel: +86-18901099861
[1]This paper is an extension work of: Liu, X., Yan, Z., Geng, G., Lee, X., Tseng, S.-S. and Ku, C.-H.. RPKI Deployment: Risks and Alternative Solutions. In the 9th International Conference on Genetic and Evolutionary Computing (ICGEC'15). Yangon, Myanmar. August 26-28 2015.

RPKI introduced many new entities and a lot of security-related data. Several challenges still need to be addressed and these include: how to produce, synchronize and use the security-related data generated by the various new entities introduced by RPKI in a safe and scalable way for large-scale deployments of RPKI in the near future.

Based on our previous work [22], we in this paper comprehensively present the potential threats associated with RPKI deployment and discuss possible solutions to counter these threats in this paper. Then we focus on the risks of the Certification Authority (CA, a key entity in RPKI) function. Based on the results obtained from empirical tests of different scenarios, we propose the CA-Safeguard scheme to secure the CA function.

The rest of the paper is organized as follows. First, we introduce the architecture of RPKI. Next, we analyze the major deployment threats of RPKI and perform some experiments about CA security threats. After identifying and evaluating the CA security threats, we propose the CA-Safeguard scheme solution to mitigate these threats. We make some concluding remarks in the last section.

2 BACKGROUND

CAs in RPKI establish the hierarchical structure which is aligned to the numerical resource (including IP addresses and Autonomous System (AS) numbers) allocation relationship [14]. Each allocated resource will be tied to a digital certificate to verify its origin property. Within the multiple certificates, the most important ones are CA certificate and End-Entity (EE) certificate: CA certificate can be used to support attestation of resource holdings and EE certificate can be used to validate the Route Origin Authorization (ROA) [21] using for verifying whether an AS is the origin of a route to a specific IP prefix [5].

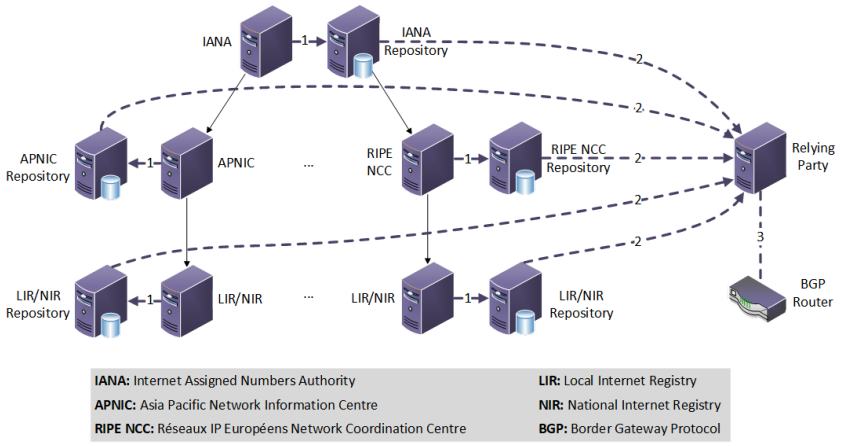

Figure 1: Verifying the route origin using RPKI

The operation of RPKI for verifying the route origin is shown in Figure 1.

- CAs publish authoritative objects (including resource certificates, ROAs and so on) into their repositories [16].
- Relying Parties[2] (RPs) collect and verify the RPKI digital objects from the repositories and store the trustful results in their local caches [27] [32].

[2]Relying Parties make use of signed objects from RPKI and also validate these objects.

- BGP routers use RP's verified results to check the route origin information contained in the received BGP message.

IETF launched the standardization of RPKI protocols and until now a series of standards have been published [12] [10] [30] [19] [25]. And the deployment of RPKI also started from Internet Assigned Numbers Authority (IANA), to five RIRs and to some countries (Ecuador, Japan, China, Bangladesh, etc.). But the adoption rate of RPKI still very low and the global deployment of global RPKI service is still in the early stage [1] [4].

3 DEPLOYMENT THREATS OF RPKI

Based on the comprehensive management of the RPKI data, we classify the RPKI threats into three categories:

3.1 Data production

In RPKI, by revoking the CA certificate, the related resource can be revoked and this operation may be caused by the unintended or malicious operations of the CA [20]. This will cause the resource holder offline [23]. In this sense, some mechanism is needed in order to prevent the unilateral revocation [11]. Besides, the signing of the digital objects may contain mistake and this can also make some resource unavailable [13].

3.2 Data synchronization

At the beginning, rsync [3] was used to synchronize the date from repositories to the RP [20]. But due to the shortcomings of rsync on security, efficiency and scalability [26] [2] [9], some other schemes have been proposed [31]. Currently, IETF has standardized the RPKI Repository Delta Protocol (RRDP) [29] to be used in stead of rsync.

3.3 Data usage

RPKI could address the problem of BGP origin validation, but it cannot protect against the path hijacking. BGPsec for the BGP path validation can do this function by signing in every step of BGP Update forwarding [25].

4 TESTS OF CA MISCONFIGURATIONS

The CA performs the most important function of RPKI and is the source of RPKI data. In this section, the scenarios and potential risks of CA operation are evaluated empirically. As described in [20], the resource holder may reallocate portions of pool of resources to the sub-nodes. But some unexpected scenarios can be caused by the misconfigurations or malicious operations of the CA as described below (all scenarios of resource allocation described in this paper apply to both IP addresses and AS numbers, and for simplicity, we only describe these scenarios with AS Number (ASN) allocation as an example).

4.1 Unauthorized resource assignment

In this scenario, a CA allocates ASNs which do not belong to it, so the sub-node cannot use those ASNs in reality. But the assignment can be conducted in RPKI. This scenario may be caused by the

misconfigurations of the CA or because of its malicious operations. Additionally, this scenario can be divided into two kinds of sub-scenarios: completely unauthorized assignment and partially unauthorized assignment.

(1) Completely unauthorized assignment: the resources to be allocated to subordinate node are without the ownership of the CA.

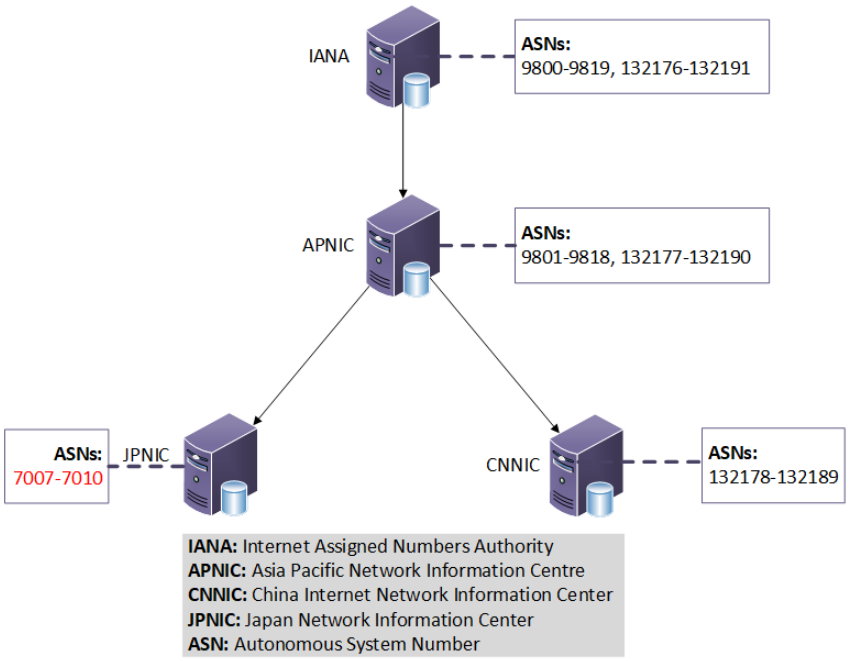

Figure 2: Scenario of a completely unauthorized assignment

The test scenario is shown in Figure 2. APNIC allocates ASNs 7007-7010 to the Japan Network Information Center (JPNIC). But APNIC does not own these ASNs completely. We simulated this process of completely unauthorized assignment on our testbed with the software of RPKI.NET which is the most common software used for the RPKI experiment and deployment. The results are illustrated in Figure 3 and Figure 4. For APNIC, it has allocated the unauthorized resources to JPNIC successfully, but for

```
root@ubuntu:~# rpkic -i apnic show_child_resources
Child: cnnic
  ASN: 132178-132189
Child: jpnic
  ASN: 7007-7010
```

Figure 3: Result of parent node in the completely unauthorized assignment

JPNIC it did not receive the requested resources.

(2) Partially unauthorized assignment: the resources to be allocated to subordinate node are partially owned by CA.

The test scenario is shown in Figure 5. APNIC allocates ASNs 9700-9818 to JPNIC. But APNIC only takes the ownership of ASNs 9801-9818, the remaining ones (ASNs 9700-9800) do not belong to APNIC. We simulated this process and the results are illustrated in Figure 6 and Figure 7. For APNIC, it has allocated the partially unauthorized ASNs to JPNIC successfully, but for JPNIC it only received the legal part (ASNs 9801-9818) without the illegal part (ASNs 9700-9800).

```
root@ubuntu:~# rpkic -i cnnic show_received_resources
Parent:      apnic
  notBefore: 2016-01-14T14:52:29Z
  notAfter:  2016-10-18T12:40:06Z
  URI:       rsync://localhost/rpki/iana/apnic/kE5ZehYmbOffKIlejVaJKYjuR-U.cer
  SIA URI:   rsync://localhost/rpki/iana/apnic/cnnic/
  AIA URI:   rsync://localhost/rpki/iana/YpF-8KpTQO_C-DfkEoFY5Emy-iA.cer
  ASN:       132178-132189
  IPv4:
  IPv6:
root@ubuntu:~# rpkic -i jpnic show_received_resources
root@ubuntu:~#
```

Figure 4: Result of child nodes in the completely unauthorized assignment

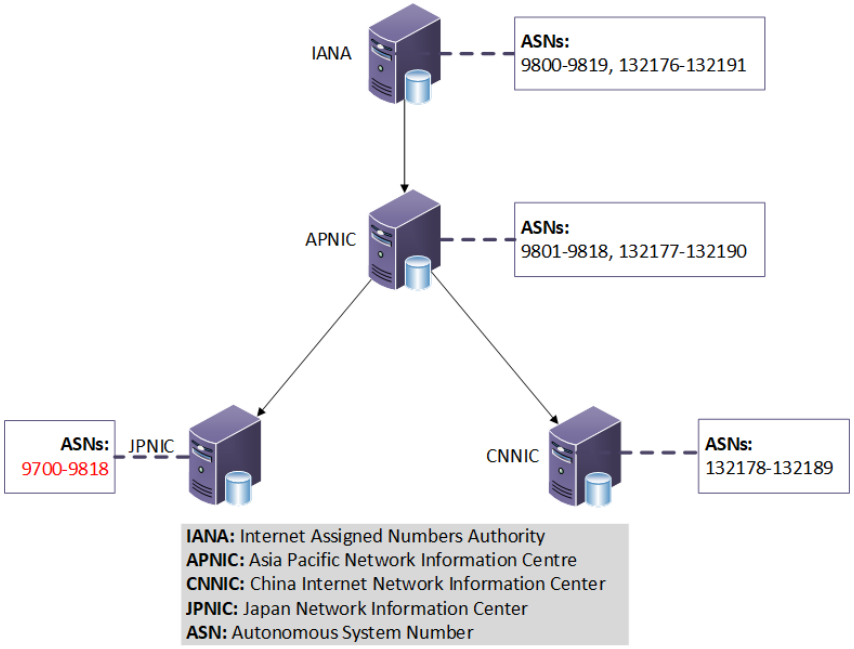

Figure 5: Scenario of partially unauthorized assignment

The above two scenarios (completely unauthorized assignment and partially unauthorized assignment) show that although the upper-layer CA (APNIC) can execute the allocation commands successfully whereas the lower-layer CAs (CNNIC and JPNIC) will not keep and own these resources in practice. Consequently, this would cause the lower-layer CAs (CNNIC and JPNIC) to be unable to use these resources. The main reason is because there is no mechanism to detect and prevent the upper-layer CA (APNIC) from allocating (accidentally or deliberately) unauthorized resources to its subordinates (CNNIC and JPNIC).

```
root@ubuntu:~# rpkic -i apnic show_child_resources
Child: cnnic
  ASN: 132178-132189
Child: jpnic
  ASN: 9700-9818
```

Figure 6: Result of parent node in the partially unauthorized assignment

```
root@ubuntu:~# rpkic -i cnnic show_received_resources
Parent:       apnic
  notBefore:  2016-01-14T14:52:29Z
  notAfter:   2016-10-18T12:40:06Z
  URI:        rsync://localhost/rpki/iana/apnic/kE5ZehYmbOffKIlejVaJKYjuR-U.cer
  SIA URI:    rsync://localhost/rpki/iana/apnic/cnnic/
  AIA URI:    rsync://localhost/rpki/iana/YpF-8KpTQO_C-DfkEoFY5Emy-iA.cer
  ASN:        132178-132189
  IPv4:
  IPv6:
root@ubuntu:~# rpkic -i jpnic show_received_resources
Parent:       apnic
  notBefore:  2016-01-14T14:56:30Z
  notAfter:   2016-10-18T12:40:10Z
  URI:        rsync://localhost/rpki/iana/apnic/1AyhgY_q_wz6YLWcSOcHRogmj2E.cer
  SIA URI:    rsync://localhost/rpki/iana/apnic/jpnic/
  AIA URI:    rsync://localhost/rpki/iana/YpF-8KpTQO_C-DfkEoFY5Emy-iA.cer
  ASN:        9801-9818
  IPv4:
  IPv6:
```

Figure 7: Result of child nodes in the partially unauthorized assignment

4.2 Resource re-assignment

In this scenario, a CA reassigns the resources which have been previously assigned to one sub-node to another sub-node. According to the resources re-assigned to different sub-nodes, this scenario could be divided into three types:

(1) Matching: the block of resources reassigned are the same as the resources which have been assigned to the other sub-node.

As shown in Figure 8, in the Matching situation, the ASNs to be allocated to JPNIC (ASNs 132178-132189) are identical to those allocated to CNNIC (ASNs 132178-132189).

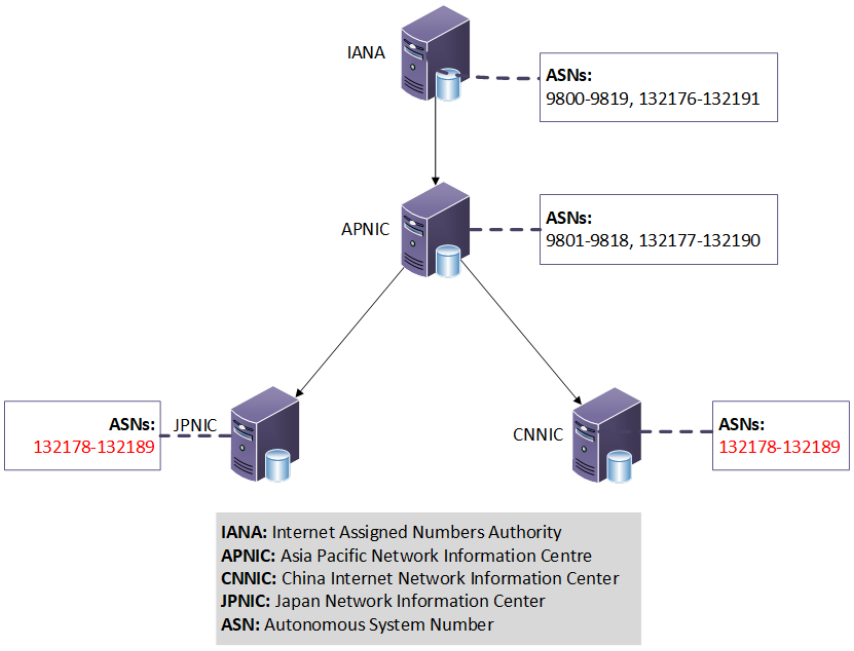

Figure 8: Scenario of matching case in resource reassignment

The test results (as shown in Figure 9 and Figure 10) show that both CNNIC and JPNIC can receive these ASNs.

```
root@ubuntu:~# rpkic -i apnic show_child_resources
Child: cnnic
  ASN: 132178-132189
Child: jpnic
  ASN: 132178-132189
```

Figure 9: Result of the parent node in the matching case for the resource re-assignment

```
root@ubuntu:~# rpkic -i cnnic show_received_resources
Parent:      apnic
  notBefore: 2016-01-14T14:52:29Z
  notAfter:  2016-10-18T12:40:06Z
  URI:       rsync://localhost/rpki/iana/apnic/kE5ZehYmbOffKIlejVaJKYjuR-U.cer
  SIA URI:   rsync://localhost/rpki/iana/apnic/cnnic/
  AIA URI:   rsync://localhost/rpki/iana/YpF-8KpTQO_C-DfkEoFY5Emy-iA.cer
  ASN:       132178-132189
  IPv4:
  IPv6:
root@ubuntu:~# rpkic -i jpnic show_received_resources
Parent:      apnic
  notBefore: 2016-01-14T15:04:32Z
  notAfter:  2016-10-18T12:40:10Z
  URI:       rsync://localhost/rpki/iana/apnic/kZmxMCVIVdxiw1Ixeg7xpVq-pjE.cer
  SIA URI:   rsync://localhost/rpki/iana/apnic/jpnic/
  AIA URI:   rsync://localhost/rpki/iana/YpF-8KpTQO_C-DfkEoFY5Emy-iA.cer
  ASN:       132178-132189
  IPv4:
  IPv6:
```

Figure 10: Result of child nodes in the matching case for the resource re-assignment

(2) Subset: the block of resources reassigned is the subset of the block of resources already assigned to the other sub-node.

As shown in Figure 11, in the Subset situation, the ASNs to be allocated to JPNIC (ASNs 132178-132180) are the subset of those allocated to CNNIC (ASNs 132178-132189).

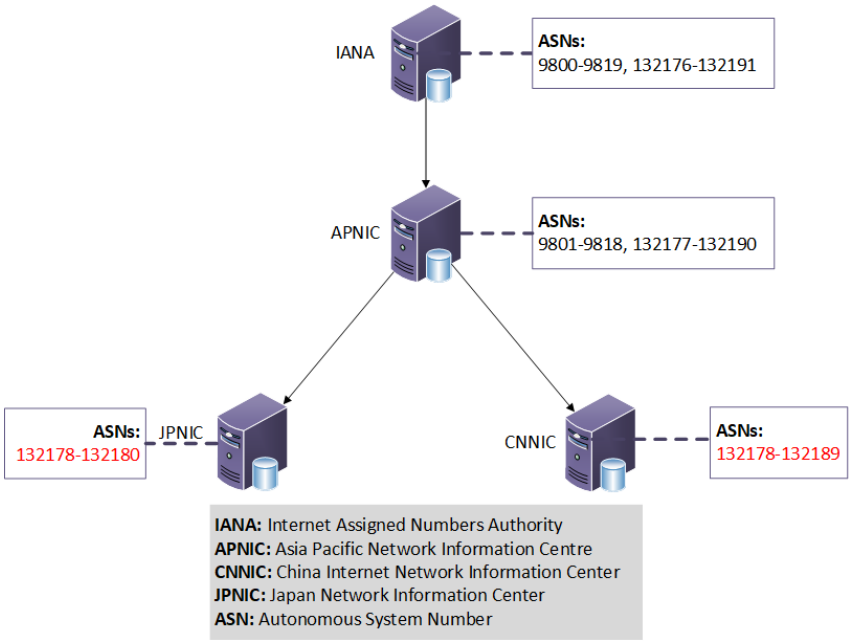

Figure 11: Scenario of subset case for the resource re-assignment

The test results (as illustrated in Figure 12 and Figure 13) show that both CNNIC and JPNIC can receive these ASNs.

```
root@ubuntu:~# rpkic -i apnic show_child_resources
Child: cnnic
  ASN: 132178-132189
Child: jpnic
  ASN: 132178-132180
```

Figure 12: Result of the parent node in the subset case for the resource re-assignment scenario

```
root@ubuntu:~# rpkic -i cnnic show_received_resources
Parent:      apnic
  notBefore: 2016-01-14T14:52:29Z
  notAfter:  2016-10-18T12:40:06Z
  URI:       rsync://localhost/rpki/iana/apnic/kE5ZehYmbOffKIlejVaJKYjuR-U.cer
  SIA URI:   rsync://localhost/rpki/iana/apnic/cnnic/
  AIA URI:   rsync://localhost/rpki/iana/YpF-8KpTQO_C-DfkEoFY5Emy-iA.cer
  ASN:       132178-132189
  IPv4:
  IPv6:
root@ubuntu:~# rpkic -i jpnic show_received_resources
Parent:      apnic
  notBefore: 2016-01-14T15:05:48Z
  notAfter:  2016-10-18T12:40:10Z
  URI:       rsync://localhost/rpki/iana/apnic/kZmxMCVIVdxiw1Ixeg7xpVq-pjE.cer
  SIA URI:   rsync://localhost/rpki/iana/apnic/jpnic/
  AIA URI:   rsync://localhost/rpki/iana/YpF-8KpTQO_C-DfkEoFY5Emy-iA.cer
  ASN:       132178-132180
  IPv4:
  IPv6:
```

Figure 13: Result of child nodes in the subset case for the resource re-assignment scenario

(3) Intersection: the block of resources reassigned has overlaps with the block of resources which have been assigned to others.

As shown in Figure 14, in the Intersection case, the ASNs to be allocated to JPNIC (ASNs 132180-132190) and those allocated to CNNIC (ASNs 132177-132185) overlap. The test results (as illustrated in Figure 15 and Figure 16) show that both CNNIC and JPNIC can receive these ASNs.

5 PROPOSED SAFEGUARD SCHEME FOR CA FUNCTION

To avoid the risks of unilateral resource revocation, Heilman et al. proposed a scheme [13] to balance the powers among the CAs in RPKI hierarchy. And S. Kent et al. also proposed the so-called ”Suspenders” [18] to address the adverse effects on INR holders which were caused by CAs’ mis-operation or malicious behavior.

These solutions are effective to regulate the CA operation but induce additional burden on RP to collect additional data. As illustrated above, RP can be used to verify the data produced by the CAs at each level. In this case, is it possible to avoid or discover the malfunctioning of the CA in practice? If so, an RP tool can be deployed to verify the RPKI data before it is finally published into the repository. We conducted the following experiment to check this assumption. We also used the above mentioned testbed, for the APNIC node, the location of its repository is:

/usr/share/rpki/publication/iana/apnic

And the location to store the verified data by RP is:

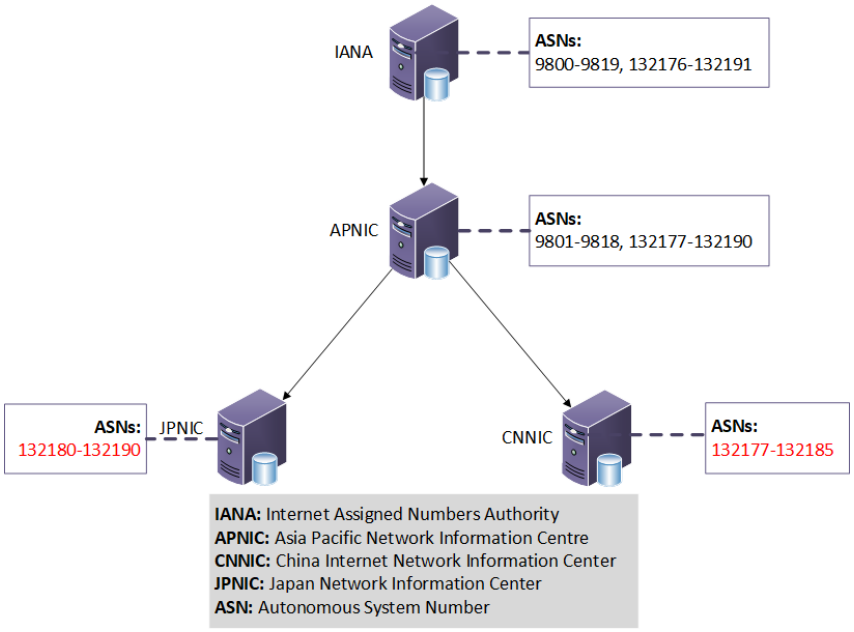

Figure 14: Scenario of intersection case for the resource re-assignment

```
root@ubuntu:~# rpkic -i apnic show_child_resources
Child: cnnic
  ASN: 132177-132185
Child: jpnic
  ASN: 132180-132190
```

Figure 15: Result of parent node in the intersection case for the resource re-assignment scenario

```
root@ubuntu:~# rpkic -i cnnic show_received_resources
Parent:      apnic
  notBefore: 2016-01-14T15:10:33Z
  notAfter:  2016-10-18T12:40:06Z
  URI:       rsync://localhost/rpki/iana/apnic/kE5ZehYmbOffKIlejVaJKYjuR-U.cer
  SIA URI:   rsync://localhost/rpki/iana/apnic/cnnic/
  AIA URI:   rsync://localhost/rpki/iana/YpF-8KpTQO_C-DfkEoFY5Emy-iA.cer
  ASN:       132177-132185
  IPv4:
  IPv6:
root@ubuntu:~# rpkic -i jpnic show_received_resources
Parent:      apnic
  notBefore: 2016-01-14T15:10:33Z
  notAfter:  2016-10-18T12:40:10Z
  URI:       rsync://localhost/rpki/iana/apnic/kZmxMCVIVdxiw1Ixeg7xpVq-pjE.cer
  SIA URI:   rsync://localhost/rpki/iana/apnic/jpnic/
  AIA URI:   rsync://localhost/rpki/iana/YpF-8KpTQO_C-DfkEoFY5Emy-iA.cer
  ASN:       132180-132190
  IPv4:
  IPv6:
```

Figure 16: Result of child nodes in the intersection case for the resource re-assignment scenario

/var/rcynic/data/authenticated/localhost/rpki/iana/apnic Since only the data that successfully passes the verification by RP could be stored in the */var/rcynic/data/authenticated/* directory, we can conclude if all the data produced by APNIC is correct by comparing the data stored in the above two directories.

```
root@ubuntu:~# ls -l /usr/share/rpki/publication/iana/apnic/
total 24
drwxr-xr-x 2 root root 4096 Jan 14 23:10 cnnic
drwxr-xr-x 2 root root 4096 Jan 14 23:10 jpnic
-rw-r--r-- 1 root root 1237 Jan 14 23:10 kE5ZehYmbOffKIlejVaJKYjuR-U.cer
-rw-r--r-- 1 root root 1237 Jan 14 23:10 kZmxMCVIVdxiw1Ixeg7xpVq-pjE.cer
-rw-r--r-- 1 root root  520 Jan 14 23:09 YpF-8KpTQO_C-DfkEoFY5Emy-iA.crl
-rw-r--r-- 1 root root 1946 Jan 14 23:10 YpF-8KpTQO_C-DfkEoFY5Emy-iA.mft
root@ubuntu:~# ls -l /var/rcynic/data/authenticated/localhost/rpki/iana/apnic/
total 24
drwxrwxr-x 2 rcynic rcynic 4096 Jan 14 23:13 cnnic
drwxrwxr-x 2 rcynic rcynic 4096 Jan 14 23:13 jpnic
-rw-rw-r-- 1 rcynic rcynic 1237 Jan 14 23:10 kE5ZehYmbOffKIlejVaJKYjuR-U.cer
-rw-rw-r-- 1 rcynic rcynic 1237 Jan 14 23:10 kZmxMCVIVdxiw1Ixeg7xpVq-pjE.cer
-rw-rw-r-- 1 rcynic rcynic  520 Jan 14 23:09 YpF-8KpTQO_C-DfkEoFY5Emy-iA.crl
-rw-rw-r-- 1 rcynic rcynic 1946 Jan 14 23:10 YpF-8KpTQO_C-DfkEoFY5Emy-iA.mft
```

Figure 17: The result of RP verification

As shown in Figure 17, unauthorized resource assignment and resource reassignment cannot be detected based on the current RP function. This means that we need to improve the RP function or propose some novel solution. In this context, S. Kent et al. proposed a mechanism based on hysteresis operation and a confirmation scheme that should be adopted by RPs to discover the CA's malfunction [17]. But this mechanism is a post-processing solution that aims to detect the malfunctioning of the CA based on the verification scheme of RP. This approach suffers from the following shortcomings:

- It is difficult to determine the time limit needed for the hysteresis and confirmation mechanism: the delay time should be set long enough to guarantee that the affected CA could recover from malfunctions or malicious attacks but it should also be short enough to avoid unnecessary delays in the processing of valid data.
- The CA needs to guarantee that the confirmation scheme is independent and secure enough which makes the CA more complex and increases its load.

Based on the above experiments and the considerations, we propose the "pre-processing" scheme, to avoid malfunctions before the certificates are wrongly produced or revoked. For this scheme to work, the following conditions should be achieved during the process of resource allocation:

1) All the resources to be allocated should belong to the CA itself in order to avoid any unauthorized resource assignment.

2) The resources that satisfy condition 1) should not be allocated to multiple different nodes in order to prevent resource reassignment.

The procedure of the proposed CA-Safeguard is illustrated in Figure 18.

Based on the procedure depicted in Figure 18, the resources should be checked before allocation in the first round: if some resources, which do not belong to this CA, exist in the .csv file (or exist in any file/database which stores the resources to be assigned. Here we use the .csv file as the example.), the warning of "Unauthorized Resources Detected" will be triggered (as shown in Figure 19). Then the CA should modify the resources to be allocated in the .csv file to avoid an unauthorized resource assignment. After this step, the second round of checking should be executed: if some resources which are repeatedly allocated to multiple sub-nodes exist in the .csv file, the warning of "Resources Re-Allocation Detected" will be triggered (as shown in Figure 20). Then the CA should also modify the resources to be allocated in the .csv file to avoid resource reassignment. After two rounds of successful checking, the required resources will be allocated and the related certificates can be produced. In this case, the scenarios

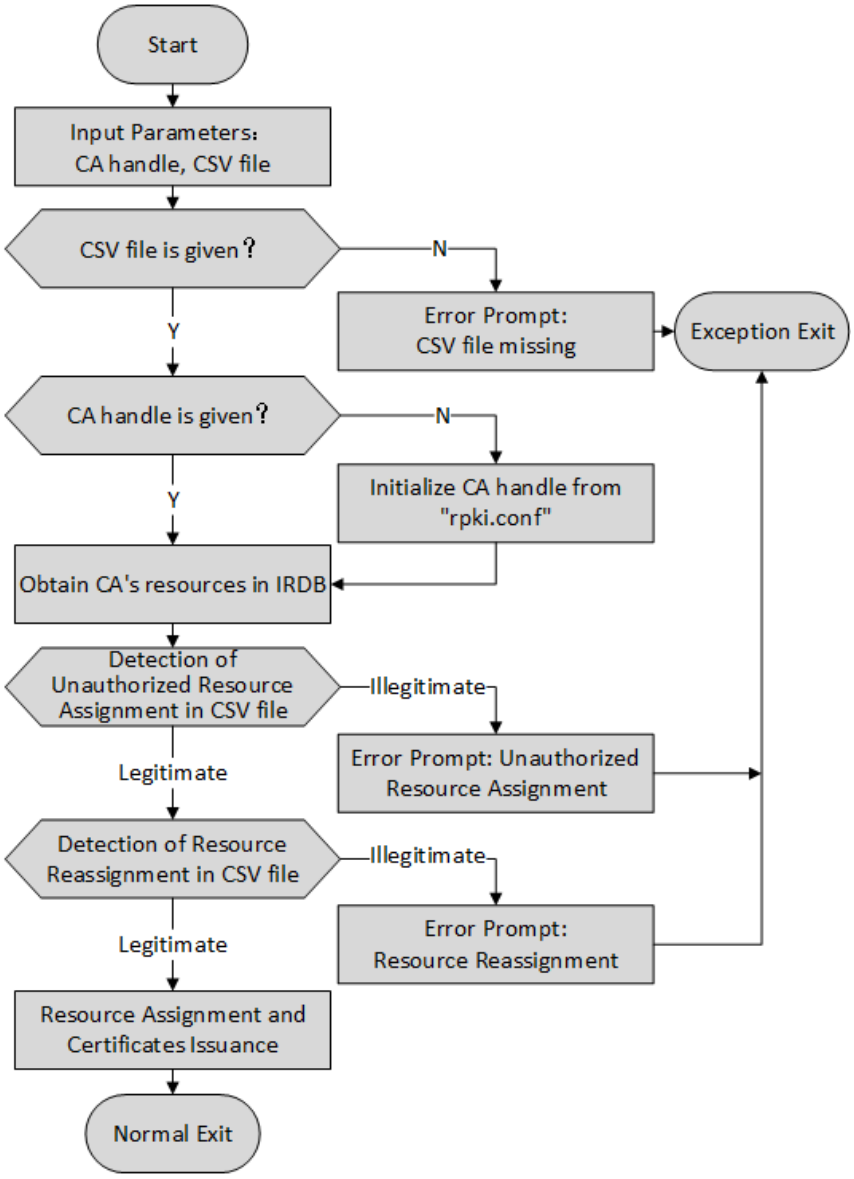

Figure 18: The proposed pre-processing procedure

```
root@ubuntu:~# load_asns_secure -i apnic apnic2Asns.csv
Unauthorized Resources Detected: apnic2Asns.csv "jpnic  7007-7010"
  AS 7007-7010 does not belong to apnic
Please modify "apnic2Asns.csv", and run "load_asns_secure" again.
```

Figure 19: Avoiding unauthorized resource assignment based on the pre-processing scheme

```
root@ubuntu:~# load_asns_secure -i apnic apnic2Asns.csv
Resources Re-Allocation Detected: apnic2Asns.csv "132178-132189"
  AS 132178-132189 is allocated more than once.
Please modify "apnic2Asns.csv", and run "load_asns_secure" again.
```

Figure 20: Avoiding resource reassignment based on the pre-processing scheme

(unauthorized resource assignment and resource re-assignment) described in Section IV can be avoided. Besides, the "pre-processing" scheme can effectively avoid the hysteresis and confirmation operations in the post-processing scheme and reduce the recovery latency and cost when the CA malfunctions.

It is worth mentioning that the safeguard scheme we proposed in this paper has taken some essential and special scenarios in RPKI such as resource transfers [6] and key rollovers [15]) into consideration. As we know, during the process of resource transfers, new certificates and ROAs for the resources need to be issued before the old ones are revoked based on the 'make-before-break' principle [24]. Additionally, during the process of key rollovers, the same resources are assigned to two different CA instances of the same CA which is performing key rollover operation. In order to be compatible with these special scenarios, the proposed safeguard scheme may provide a resource and a CA whitelist to ensure that only the resources during the process of resource transfers and key rollovers are allowed to be assigned to the different corresponding CA instances. Furthermore, to prevent accidental misconfigurations and deliberate attacks, the whitelist must be generated automatically (for example, during the process of resource transfers). The whitelist could be generated by extracting the IP prefixes, AS numbers and the corresponding CA instances specified in the Transfer Authorization Object (TAO) [8]).

6 CONCLUSION

RPKI is an important infrastructure that promises to enhance the security of BGP. RPKI uses several vital functions and introduces a lot of security-related data (CA certificates, ROAs, Manifest, etc.). In order to guarantee that these functions work smoothly and the entire architecture is scalable enough for large-scale deployment, we analyzed the security threats of RPKI from different perspectives and focused on the risks of the CA. Based on the empirical tests conducted on our testbed, we proposed a CA-Safeguard scheme and using our implementation we demonstrated its efficiency. In the next step, we will incorporate the CA-Safeguard in the RPKI deployment of CNNIC and at the same time push for its adoption in the global deployment of RPKI.

ACKNOWLEDGEMENTS

This paper was supported by the National Natural Science Foundation of China under Grant No. 61303242.

References

[1] Rpki dashboard. `http://rpki.surfnet.nl/global.html` [Online; Accessed on February 22, 2018].

[2] rsync considered inefficient and harmful. `https://www.ietf.org/proceedings/89/slides/slides-89-sidr-6.pdf` [Online; Accessed on February 22, 2018].

[3] rsync web pages. `https://rsync.samba.org` [Online; Accessed on February 22, 2018].

[4] Statistics of Resource Certification (RPKI). `http://certification-stats.ripe.net` [Online; Accessed on February 22, 2018].

[5] R. Austein, G. Huston, S. Kent, and M. Lepinski. Manifests for the Resource Public Key Infrastructure (RPKI). IETF RFC 6486, February 2012. `https://tools.ietf.org/html/rfc6486` [Online; Accessed on February 22, 2018].

[6] R. Austin, R. Bush, G. Huston, and G. Michaelson. Resource Transfer in the Resource Public Key Infrastructure. draft-ymbk-sidr-transfer-01, July 2015. `https://tools.ietf.org/html/draft-ymbk-sidr-transfer-01` [Online; Accessed on February 22, 2018].

[7] H. Ballani, P. Francis, and X. Zhang. A Study of Prefix Hijacking and Interception in the Internet. In *Proc. of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'07), Kyoto, Japan*, pages 265–276. ACM, August 2007.

[8] E. Barnes. Resource Public Key Infrastructure (RPKI) Resource Transfer Protocol and Transfer Authorization Object (TAO). draft-barnes-sidr-tao-00, February 2014. `https://tools.ietf.org/html/draft-barnes-sidr-tao-00` [Online; Accessed on February 22, 2018].

[9] W. Bryan. RPKI Repository Distribution Protocol (RRDP). `https://www.ietf.org/mail-archive/web/sidr/current/msg05367.html` [Online; Accessed on February 22, 2018].

[10] R. Bush and R. Austein. The Resource Public Key Infrastructure (RPKI) to Router Protocol. IETF RFC 6810, January 2013. `https://tools.ietf.org/html/rfc6810` [Online; Accessed on February 22, 2018].

[11] D. Cooper, E. Heilman, K. Brogle, L. Reyzin, and S. Goldberg. On the Risk of Misbehaving RPKI Authorities. In *Proc. of the 12th ACM Workshop on Hot Topics in Networks(HotNets'13), College Park, Maryland, USA*, number 16. ACM, November 2013.

[12] R. Gagliano, S. Turner, and S. Kent. Algorithm Agility Procedure for the Resource Public Key Infrastructure (RPKI). IETF RFC 6916, April 2013. `https://tools.ietf.org/html/rfc6916` [Online; Accessed on February 22, 2018].

[13] E. Heilman, D. Cooper, L. Reyzin, and S. Goldberg. From the Consent of the Routed-Improving the Transparency of the RPKI. In *Proc. of the 2014 ACM conference on SIGCOMM(SIGCOMM'14), Chicago, Illinois, USA*, pages 51–62. ACM, August 2014.

[14] G. Huston and G. Michaelson. Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs). IETF RFC 6483, February 2012. `https://tools.ietf.org/html/rfc6483` [Online; Accessed on February 22, 2018].

[15] G. Huston, G. Michaelson, and S. Kent. Certification Authority (CA) Key Rollover in the Resource Public Key Infrastructure (RPKI). IETF RFC 6489, February 2012. `https://tools.ietf.org/html/rfc6489` [Online; Accessed on February 22, 2018].

[16] G. Huston, G. Michaelson, and R. Loomans. A Profile for Resource Certficate Repository Structure. IETF RFC 6481, February 2012. `https://tools.ietf.org/html/rfc6481` [Online; Accessed on February 22, 2018].

[17] S. Kent and D. Ma. Adverse Actions by a Certification Authority (CA) or Repository Manager in the Resource Public Key Infrastructure (RPKI). IETF RFC 8211, September 2017. `https://tools.ietf.org/html/rfc8211` [Online; Accessed on February 22, 2018].

[18] S. Kent and D. Mandelberg. Suspenders: A Fail-safe Mechanism for the RPKI. draft-kent-sidr-suspenders-04, October 2015. `https://tools.ietf.org/html/draft-kent-sidr-suspenders-04` [Online; Accessed on February 22, 2018].

[19] R. Kisteleki and B. Haberman. Securing Routing Policy Specification Language (RPSL) Objects with Resource Public Key Infrastructure (RPKI) Signatures. IETF RFC 7909, June 2016. `https://tools.ietf.org/html/rfc7909` [Online; Accessed on February 22, 2018].

[20] M. Lepinski, R. Barnes, and S. Kent. An Infrastructure to Support Secure Internet Routing. IETF RFC 6480, February 2012. `https://tools.ietf.org/html/rfc6480` [Online; Accessed on February 22, 2018].

[21] M. Lepinski, D. Kong, and S. Kent. A Profile for Route Origin Authorizations (ROAs). IETF RFC 6482, February 2012. `https://tools.ietf.org/html/rfc6482` [Online; Accessed on February 22, 2018].

[22] X. Liu, Z. Yan, G. Geng, X. Lee, S.-S. Tseng, and C.-H. Ku. RPKI Deployment: Risks and Alternative Solutions. In *Proc. of the 9th International Conference on Genetic and Evolutionary Computing, Yangon, Myanmar*, volume 1, pages 299–310. Springer, Cham, August 2015.

[23] A. Malhotra and S. Goldberg. RPKI vs ROVER Comparing the Risks of BGP Security Solutions. In *Proc. of the 2014 ACM conference on SIGCOMM (SIGCOMM'14), Chicago, Illinois, USA*, pages 113–114. ACM, August 2014.

[24] T. Manderson, K. Sriram, and R. White. Use Cases and Interpretations of Resource Public Key Infrastructure (RPKI) Objects for Issuers and Relying Parties. IETF RFC 6907, March 2013. `https://tools.ietf.org/html/rfc6907` [Online; Accessed on February 22, 2018].

[25] L. Matthew and S. Kotikalapudi. BGPsec Protocol Specification. IETF RFC 8205, September 2017. `https:`

//tools.ietf.org/html/rfc8205 [Online; Accessed on February 22, 2018].

[26] M. Oleg. RPKI Repository Analysis and Delta Protocol. http://www.ietf.org/proceedings/86/slides/slides-86-sidr-2.pdf [Online; Accessed on February 22, 2018].

[27] Rcynic. http://trac.rpki.net/wiki/doc/RPKI/RP [Online; Accessed on February 22, 2018].

[28] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). IETF RFC 4271, January 2006. https://tools.ietf.org/html/rfc4271 [Online; Accessed on February 22, 2018].

[29] B. Tim, M. Oleg, and W. Bryan. RPKI Repository Analysis and Requirements, 2013 February. https://datatracker.ietf.org/doc/draft-tbruijnzeels-sidr-repo-analysis/ [Online; Accessed on February 22, 2018].

[30] B. Tim, M. Oleg, W. Bryan, and A. Rob. The RPKI Repository Delta Protocol (RRDP). IETF RFC 8182, July 2017. https://tools.ietf.org/html/rfc8182 [Online; Accessed on February 22, 2018].

[31] C. Wang, Z. Yan, and A. Hu. An Efficient Data Management Architecture for the Large-scale Deployment of Resource Public Key Infrastructure. In *Proc. of the 4th International Conference on Electronics, Communications and Networks(CECNet'14), Beijing, China*, December 2014.

[32] S. Weiler, D. Ward, and R. Housley. The rsync URI Scheme. IETF RFC 5781, February 2010. https://tools.ietf.org/html/rfc5781 [Online; Accessed on February 22, 2018].

[33] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao. Practical Defenses Against BGP Prefix Hijacking. In *Proc. of the 2007 ACM CoNext conference (CoNEXT'07), New York, New York, USA*. ACM, December 2007.

[34] J. Zhao and Y. Wen. Evaluation on the Influence of Internet Prefix Hijacking Events. *Computer Science and Information Systems*, 10(2):611–631, April 2013.

Author Biography

Zhiwei Yan received his Ph.D. degree from National Engineering Laboratory for Next Generation Internet Interconnection Devices at Beijing Jiaotong University. He joined Chinese Academy of Sciences in 2011 and is currently an Associate Professor of China Internet Network Information Center. Since April 2013, he has been an Invited Researcher of Waseda University. His research interests include mobility management, network security, and next generation Internet.

Guanggang Geng received the Ph.D. degree from the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He joined the Computer Network Information Center, Chinese Academy of Sciences, Beijing, in 2008. He is currently an Associate Professor in the China Internet Network Information Center. His current research interests include machine learning, adversarial information retrieval on the Web, and Web search.

Hidenori Nakazato received his B.Eng. degree in electronics and telecommunications from Waseda University in 1982 and his M.S. and Ph.D. degrees in computer science from the University of Illinois in 1989 and 1993, respectively. He was with Oki Electric from 1982 to 2000. Since 2000, he has been a faculty member at Waseda University, Japan. His research interests include performance issues in distributed systems and networks. He is a member of IEEE, ACM, IEICE, and IPSJ.

Yong-Jin Park received B.E., M.E., and Ph.D. degrees in Electronic Engineering from Waseda University. From 1978 to 2010, he was a Professor at Hanyang University, Seoul. He was a Visiting Associate Professor from 1983 to 1984 in the Department of Computer Science, University of Illinois, Urbana-Champaign. He was a Research Fellow at the Computing Laboratory, University of Kent, Canterbury, England from 1990 to 1991. He was the President of the Open Systems Interconnection Association from 1991 to 1992, the Chairman of the IEEE Seoul Section from 1999 to 2000, and the Director of the Secretariat of the Asia Pacific Advanced Network (APAN) from 1999 to 2003. He was the President of the Korea Institute of Information Scientists and Engineers (KIISE) in 2003 and the Director of IEEE Region 10 from 2009 to 2010. He was also a Professor of Waseda University, Tokyo, from 2010 to 2016. He joined University Malaysia Sabah in September 2016, where he is a Professor of Faculty of Computing and Informatics. Currently, he is a Professor Emeritus at Hanyang University.