CableLabs®
Security

Resource Public Key Infrastructure (RPKI) Deployment Best Common Practice

CL-GL-RPKI-BCP-V01-220120

RELEASED

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provide any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, infringement, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	CL-GL-RPKI-BCP-V01-220120			
Document Title:	Resource Public Key Infrastructure (RPKI) Deployment Best Common Practice			
Revision History:	V01 – Released 01/20/22			
Date:	January 20, 2022			
Status:	~~Work in Progress~~	~~Draft~~	**Issued**	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document that is considered largely complete, but lacking review by members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A generally public document that has undergone rigorous member and technology supplier review, cross-vendor interoperability, and is suitable for certification/qualification testing if applicable.
Closed	A static document, reviewed, tested, validated, and closed to further changes.

Trademarks

Contents

Figures

1 SCOPE

1.1 Introduction and Overview

The Internet consists of more than 70,000 autonomous systems as of June 2, 2021. Each autonomous system (AS), identified by a unique autonomous system number (ASN), implements a single and clearly defined routing policy for a collection of IP address spaces, namely IP prefixes [RFC 1930]. The Border Gateway Protocol (BGP) [RFC 4271] is the Internet Engineering Task Force (IETF) standard interdomain routing protocol for exchanging routing information between one AS and another. An AS announces its IP prefixes via a BGP update to its direct neighbors, which may further propagate the update to their respective neighbors. Each AS also appends its ASN to the AS_PATH attribute of a route when it advertises that route. A remote AS usually receives a route to a given IP prefix from each of its neighbors and follows a route selection process to select the best route. The AS uses the selected route to forward traffic destined to the address space specified by the IP prefix.

One critical security flaw with BGP is that it assumes each AS is trustworthy and does not provide any built-in mechanism to verify if an AS has proper rights to announce a given IP prefix. This opens a serious security hole that allows one AS to announce IP prefixes assigned to any other AS. This is commonly referred to as prefix hijacking, which could occur intentionally (e.g., by an attacker) or unintentionally (e.g., by misconfiguration).

There are two types of prefix hijacking: (1) announcing an IP prefix as being the same as the one announced by the legitimate AS and (2) announcing an IP prefix that is more specific than the one announced by the legitimate AS. Examples of consequences from IP prefix hijacking include but are not limited to denial of services (i.e., legitimate user traffic cannot get to its ultimate destination) and man-in-the-middle attacks (i.e., legitimate user traffic is forwarded through a router under the control of an adversary).

BGP prefix hijacking has been extensively studied by both academia and industrial practitioners since the first publicized incident in 1997. Two main academic proposals on mitigating prefix hijacking are Secure BGP (sBGP) [Kent et al.] and Pretty Secure BGP (psBGP) [Wan et al.], which use centralized and decentralized trust models, respectively.

In sBGP, two centralized public key infrastructures (PKIs), both of which are rooted at the Internet Corporation for Assigned Names and Numbers (ICANN), are used to certify the ownership of ASNs and IP prefixes, respectively. Certificates for ASNs and IP prefixes are distributed to BGP speakers using out-of-band mechanisms, and route attestation (digitally signed routes) is distributed using in-band mechanisms (i.e., within BGP updates). BGP speakers use ASN and IP prefix certificates to verify the digitally signed routes in BGP updates.

In psBGP, a centralized PKI rooted at ICANN is used to certify the ownership of ASNs. To reduce the overhead of maintaining a global PKI for IP prefix assignment, psBGP proposes to have the ownership of IP prefixes endorsed by direct neighbors. Because an AS can obtain the knowledge of IP prefixes owned by a direct neighbor during the establishment of their peering agreement, its endorsement improves confidence in the correctness of routes announced by its neighbors, albeit not in the authoritative capacity. This is similar to a social trust model, in which trust can be established on an individual based on the endorsement of the person's references in the absence of ground truth. Both sBGP and psBGP require BGP speakers to perform cryptographic operations, which makes them difficult to deploy.

In 2006, the IETF started the Secure Inter-Domain Routing (SIDR) working group to standardize resource public key infrastructure (RPKI). The first Internet draft of RPKI was submitted in 2007 and became [RFC 6480] in 2012. RPKI is based on the centralized hierarchical trust model used by sBGP but differs from sBGP in two aspects. (1) RPKI is rooted at each of the five regional Internet registries (RIRs), whereas sBGP is rooted at ICANN. (2) RPKI does not require BGP speakers to perform cryptographic operations because they are performed by an RPKI validator located outside of the BGP speaker.

Four of the RIRs started to offer RPKI services in January 2011; the fifth, the American Registry for Internet Numbers (ARIN), did so in September 2012 [Chung et al.]. However, the deployment of RPKI by operators had a slow start because of a lack of motivation—an early adopter does not receive much benefit unless many other operators also deploy it. The NIST RPKI monitor [NIST] shows that in October 2018, about seven years after RPKI services became available, only the first 10% of global unique IP prefix and origin ASN pairs had been validated in RPKI. However, RPKI deployment accelerated in 2019, so the amount of pairs protected by RPKI increased to 20% in under two years (May 2020). In another year (June 2021), 30% of pairs had been protected.

1.2 Purpose of Document

An operator choosing to deploy RPKI now can immediately benefit because of the accelerated deployment of RPKI on the Internet by other operators. The spread of RPKI deployment can help stop the hijacking of prefixes if they are digitally signed and published. However, the deployment of RPKI requires planning and experience; otherwise, it may encounter obstacles and introduce new risks to networks.

To help network engineers deploy RPKI, we developed this RPKI deployment best common practice with a focus on route origin authorization (ROA) [RFC 6482] and route origin validation (ROV) deployment, based on the firsthand experience of the co-authors, who have successfully deployed RPKI in their networks. Note that these recommendations are based on the deployment of RPKI in cable networks, so they may need to be adjusted when applied to other types of networks. This document does not cover other RPKI-related solutions such as BGPsec and autonomous system provider authorization (ASPA).

2 REFERENCES

2.1 Informative References

[Chung et al.] "RPKI is Coming of Age: A Longitudinal Study of RPKI Deployment and Invalid Route Origins," T. Chung et al., ACM Internet Measurement Conference, October 21–23, 2019, Amsterdam, Netherlands.

[Gilad et al.] "Are We There Yet? On RPKI's Deployment and Security," Y. Gilad, A. Cohen, A. Herzberg, M. Schapira, H. Shulman, NDSS Symposium (Network and Distributed System Security), February 26–March 1, 2017, San Diego, CA.

[Kent et al.] "Secure Border Gateway Protocol (S-BGP)," S. Kent, C. Lynn, K. Seo, IEEE Journal on Selected Areas in Communications, v. 18, no. 4, April 2000.

[maxLength] IETF draft-ietf-sidrops-rpkimaxlen-09, The Use of maxLength in the RPKI, Y. Gilad, S. Goldberg, K. Sriram, J. Snijders, B. Maddison, November 2021.

[NIST] NIST RPKI Monitor, https://rpki-monitor.antd.nist.gov/.

[RFC 1930] IETF RFC 1930, Guidelines for Creation, Selection, and Registration of an Autonomous System (AS), J. Hawkinson, T. Bates, March 1996.

[RFC 2827] IETF RFC 2827 (BCP 38), Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing, P. Ferguson, D. Senie, May 2000.

[RFC 3013] IETF RFC 3013 (BCP 46), Recommended Internet Service Provider Security Services and Procedures, T. Killalea, November 2000.

[RFC 4271] IETF RFC 4271, A Border Gateway Protocol 4 (BGP-4), Y. Rekhter, T. Li, S. Hares, January 2006.

[RFC 6480] IETF RFC 6480, An Infrastructure to Support Secure Internet Routing, M. Lepinski, S. Kent, February 2012.

[RFC 6482] IETF RFC 6482, A Profile for Route Origin Authorizations (ROAs), M. Lepinski, S. Kent, D. Kong, February 2012.

[RFC 6486] IETF RFC 6486, Manifests for the Resource Public Key Infrastructure (RPKI), R. Austein, G. Huston, S. Kent, M. Lepinski, February 2012.

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

[RFC 6811] IETF RFC 6811, BGP Prefix Origin Validation, P. Mohapatra et al., January 2013.

[RFC 7908] IETF RFC 7908, Problem Definition and Classification of BGP Route Leaks, K. Sriram et al., June 2016.

[RFC 8182] IETF RFC 8182, The RPKI Repository Delta Protocol (RRDP), T. Bruijnzeels, O. Muravskiy, B. Weber, R. Austein, July 2017.

[Wan et al.] "Pretty Secure BGP (psBGP)," T. Wan, E. Kranakis, P.C. van Oorschot, NDSS Symposium (Network and Distributed System Security), February 2005, San Diego, CA.

2.2 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone: +1-303-661-9100; Fax: +1-303-661-9199; http://www.cablelabs.com
- IETF Secretariat, c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434; Phone: +1-703-620-8990; Fax: +1-703-620-9071; http://www.ietf.org

3 TERMS AND DEFINITIONS

It is assumed that readers are familiar with terms relevant to BGP and RPKI, such as autonomous system (AS), autonomous system number (ASN), public key infrastructure (PKI), certification authority (CA), and X.509v3 certificate.

4 ABBREVIATIONS AND ACRONYMS

This document uses the following abbreviations.

AFRINIC	African Network Information Centre
API	application programming interface
APNIC	Asia Pacific Network Information Centre
ARIN	American Registry for Internet Numbers
AS	autonomous system
ASN	autonomous system number
ASPA	autonomous system provider authorization
BGP	Border Gateway Protocol
BMP	BGP Monitoring Protocol
CA	certificate authority
CLI	command line interface
CMS	cryptographic message syntax
CNNIC	China Internet Network Information Center
CRL	certificate revocation list
DDoS	distributed denial-of-service
EE	end entity
HA	high availability
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ICANN	Internet Corporation for Assigned Names and Numbers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPAM	IP address management
ISOC	Internet Society
ISP	Internet service provider
JPNIC	Japan Network Information Center
LACNIC	Latin America and Caribbean Network Information Centre
NTT	NTT Communications
OV	origin validation
PA	provider aggregatable
PKI	public key infrastructure
RIPE	Réseaux IP Européens (European IP Networks)
RIR	regional Internet registry
RIS	routing information service
RP	relying party
RPKI	resource public key infrastructure
ROA	route origin authorization
ROV	route origin validation
RRDP	RPKI Repository Delta Protocol
RTBH	real-time black hole
RTR	RPKI to router

SIDR	Secure Inter-Domain Routing
TAL	trust anchor list
TWNIC	Taiwan Network Information Center
VC	validating cache
VRP	validated ROA payload

5 OVERVIEW OF RPKI

The resource public key infrastructure (RPKI) [RFC 6480] is an out-of-band cryptography framework that enhances security for the interdomain routed network. In particular, it enables network operators to assert ownership of their IP number resources and validate the legitimacy of incoming BGP route advertisements from external peers. RPKI provides an IP prefix to ASN mapping by binding an IP prefix to the ASN that is authorized to originate it. To accomplish this, the RPKI leverages the use of X.509 PKI certificates with relevant extensions for IP prefixes and ASNs. This practice, once adopted, effectively mitigates the risk of BGP hijacks relating to origin attacks or some origin-related route leaks, which, as specified by [RFC 7908], occurs when an AS inadvertently announces a prefix that violates the policies of both the sender and receiver and results in an unintentional redirection of traffic.

The foundation of RPKI for BGP security is a certification authority (CA) with a hierarchical trust model in which an authoritative entity issues digital certificates to convey ownership of IP resources. Because the structure and primary functions of a CA resemble IP number assignment by the Internet Assigned Numbers Authority (IANA) and the RIRs, the RIRs are rightfully suited to serve in the role of certificate authorities for their respective geographical areas. The five RIRs hold the root certificates for all IP prefixes and ASNs.

As IP prefixes and ASNs are assigned to Internet service providers (ISPs), a corresponding end entity (EE) certificate (or resource certificate) can also be issued to digitally represent these prefixes within the RPKI ecosystem. An EE certificate is an X.509 digital certificate with important extensions for IP number resources. ISP administrators will need to explicitly request resource certificates for their IP number assignments. When the requests are made, either by email or through the use of an application programming interface (API), the network administrator must include its organization's public key. The ensuing resource certificate contains several fields, but the specific IP prefix assignment for the ISP, the ISP's public key, and the issuing CA's public key certificate identifier are of the utmost importance. The certificate represents proof of ownership for the ISP for the IP resources specified. Figure 1 illustrates the assignment of a resource certificate for prefix y/16 from the ARIN to ISP-3.

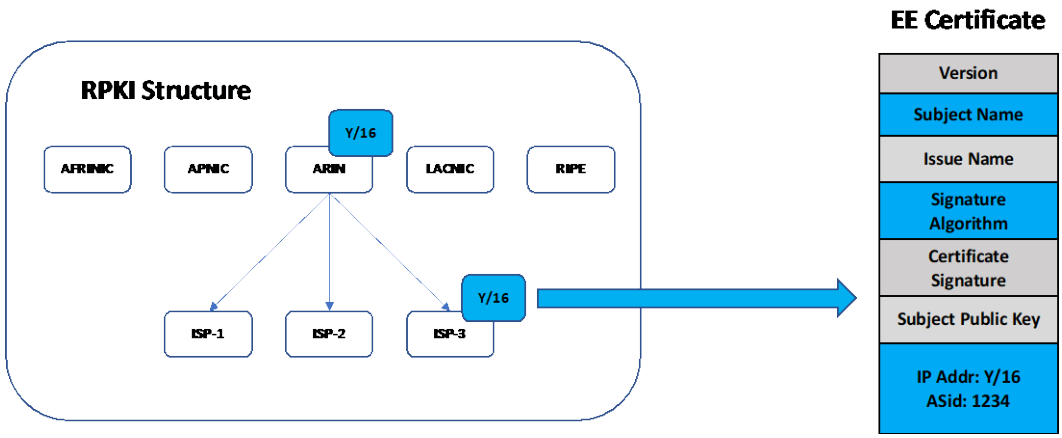

Figure 1 - RPKI Certificate Authority Structure
[This resource certificate has been abbreviated for purposes of this illustration.]

After an ISP has received a resource certificate from an RIR, it can authorize autonomous systems to originate IP prefixes specified in the resource certificate. This task is completed within the CA structure, and these authorizations are referred to as route origin authorizations (ROAs, pronounced *ROW-ahs*). ROAs are digitally signed objects based on cryptographic message syntax (CMS) format. When creating ROAs, network administrators will specify the following information:

- one or more IP prefixes,
- a single ASN that is allowed to originate these prefixes,
- an optional field called maxLength,
- the date when the ROA will become valid (usually when the ROA was created), and
- the date when the ROA will expire.

More importantly, the network operator's EE certificate is added to the ROA, allowing any other party to verify that the claimed IP prefixes in the ROA are contained in the IP prefixes in the EE certificate. The network operator's EE certificate is verified to chain back to a trusted RIR root certificate. In Figure 2, ISP-3 creates a ROA to authorize AS 1234 to originate prefix Y/16 with a max length of /24.

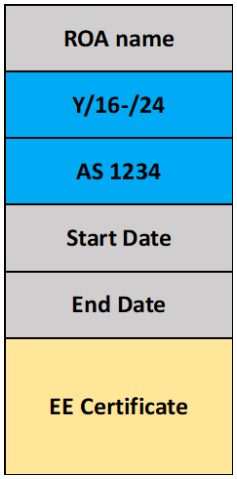

Figure 2 - An Example of a ROA

The RPKI CA uses a publication server to make objects such as ROAs, manifests, and certificate revocation lists (CRLs) available to relying parties (RPs) needing to verify this information. RPs can use the rsync protocol to connect to this server and download RPKI data. Specifically, RPs use a validating cache (VC) to retrieve these objects from the publication server. The VC will verify the legitimacy of all signed objects and produce a validated ROA payload (VRP)—an IP prefix to the ASN database containing the IP prefix, maximum length, and originating AS. This VRP database is made accessible to the ISP's edge routers using the RPKI to router protocol per [RFC 6810]. Here, the router issues a request to the VC and receives all of its data records. The edge routers then use the BGP route origin validation (ROV) protocol per [RFC 6811] to store the VRPs in its route validation database. By using ROV, edge routers can assign a validation state for each incoming BGP prefix in the update message. The possible validation states, per [RFC 6811], are valid, invalid, and unknown, as described below.

- Valid—There is a VRP for the prefix, and it matches.
- Invalid—There is a VRP for the prefix, but it does not match.
- Unknown—There is no VRP for the prefix.

The ISP can use these validation states to create policies that filter or drop all invalid routes to protect the network against BGP hijacks.

Figure 3 illustrates the communications between the CA's publication server, the VC, and the RP's network.

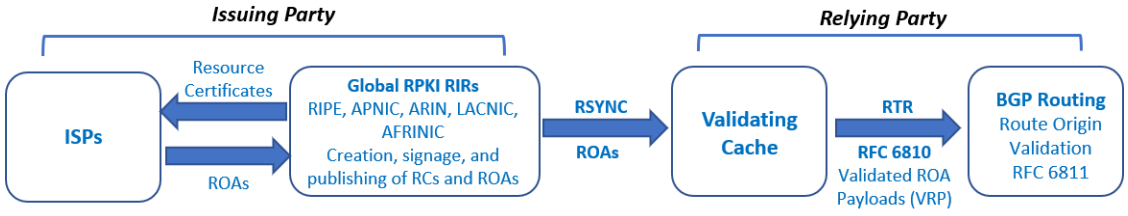

Figure 3 - RPKI High-Level Workflow

6 DEPLOYMENT OF ROUTE ORIGIN AUTHORIZATIONS (ROAS)

Implementing ROAs on a network involves five high-level steps.

1. Identify the "source of truth."
2. Identify the deployment model.
3. Set up the infrastructure.
4. Generate the ROAs.
5. Monitor for invalid routes.

6.1 Identify the "Source of Truth"

When creating new ROAs, it is important to first identify the "source of truth" for the creation of the ROAs. There are at least two sources of truth to be considered—the IP address management (IPAM) database and the BGP advertisements that are seen from outside an ASN or another source. Defining these sources of truth will aid in the manual or automated creation of new ROAs when new prefixes are advertised or modified.

6.2 Identify the Deployment Model

There are three possible RPKI deployment models.

- Hosted RPKI model—The RIR runs the CA, the publication server, and the repository. This model is the least expensive because the ISP has the least responsibility as well as the least control.
- Delegated RPKI model—The ISP runs everything, including the CA, the publication server, and the repository. This model is the most expensive because the ISP has all the responsibility as well as all the control.
- Hybrid RPKI model—The ISP runs the CA, but the RIR runs the publication server and the repository. This model is balanced, giving the ISP only some of the control and responsibility.

Figure 4 is a flowchart that can help one select a suitable deployment model.

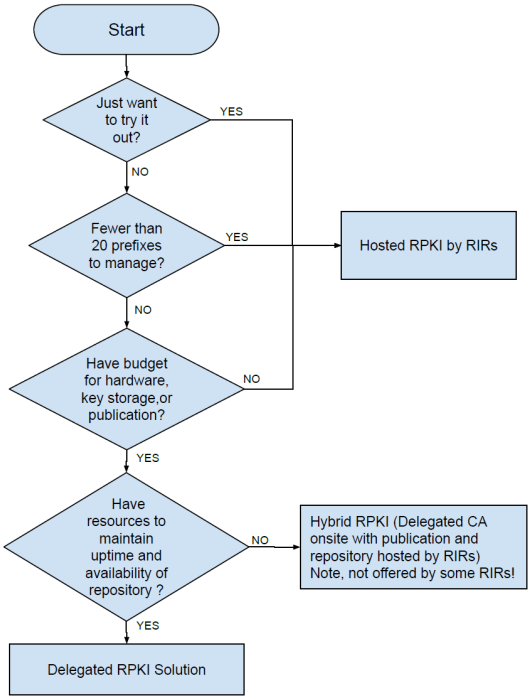

Figure 4 - Process for ROA Deployment Model Selection

6.2.1 Hosted RPKI Model

In the hosted RPKI model, the RIR is responsible for maintaining the CA, publication server, and repository. The ISP is responsible for the creation and deletion of ROAs.

This model has the following advantages.

- It is very easy for end users to implement because the RIR is responsible for the entire solution.
- It is free; there is no additional cost beyond leasing prefixes from the RIR.
- ARIN, RIPE, and other RIRs have an API for generating, listing, and deleting ROAs.
- It transfers the burden of maintaining a CA, publication server, and repository to the RIR.

This model has the following disadvantages.

- The RIR APIs have limited functionality compared to solutions like Krill.
- The RIRs have had some issues with the availability of their repositories.
- The ROA certificates do not automatically renew.
- The RIR interfaces may not be user friendly.
- It does not offer suggestions for easy ROA creation.
- It requires the ORG ID to match with the prefix.
- It is easily susceptible to fat fingering or incorrectly pasting data, which can cause RPKI INVALID route announcements.

6.2.2 Delegated RPKI Model

In the delegated RPKI model, prefix holders run their own CA (the delegated CA), publication server, and repository. Krill by NLnet Labs (https://nlnetlabs.nl/projects/rpki/krill/) is a popular open source project used for this model.

This model has the following advantages.

- Krill has a fully functional API.
- It can provide better integration with existing infrastructure (e.g., integrate with IPAM and monitoring).
- It has better security (the private key stays in the prefix holder's control).
- Krill provides automatic renewal of expiring certificates.
- Krill shows how a potential ROA will affect a prefix holder's advertisement before creation.
- It allows the prefix holder to manage resources from multiple RIRs in a single instance.
- It allows the prefix holder to delegate a subset of resources to a different business unit or a customer so that they can manage ROAs themselves.

This model has the following disadvantages.

- The prefix holder is responsible for making sure the system is up and available. This includes maintaining redundancy, HA, scalability, protection from DDoS attacks, etc.
- It costs money, time, and manpower.
- Krill is the only delegated CA implementation that is actively maintained. The RPKI Toolkit from Dragon Research Labs (https://github.com/dragonresearch/rpki.net), used by TWNIC, CNNIC, and JPNIC to run their CAs, does not appear to be actively maintained.

6.2.3 Hybrid RPKI Model

The hybrid RPKI model is a hybrid solution in that the ISP runs a delegated CA but the RIR runs the publication server and repository. Currently, this option appears to only be available with APNIC. RIPE, ARIN, and LACNIC may offer this option in the future.

This model has the following advantages.

- The RIR is responsible for making sure the repository is up and available, so this risk is removed from the prefix holder.
- It has all of the other advantages of the delegated CA outlined in the delegated RPKI model.

This model has the following disadvantages.

- RIRs have had some issues with keeping up their repositories.
- The prefix holder will have to stand up and maintain a CA.

6.3 Set up the Infrastructure

The next step is for the ISP to set up the infrastructure, including the servers, networking, storage, and connections to API (for IPAM). If the delegated RPKI deployment model is used, load balancing and DDoS detection and/or mitigation also must be set up. Further, scripts for monitoring the ROAs may need to be developed.

6.4 Generate the ROAs

When generating ROAs, one ROA should be generated for each prefix. ROAs might also need to be generated for the prefixes that are not yet advertised to prevent prefix squatting.

As stated in a previous section, a ROA can contain more than one prefix. Only one prefix should be put into a ROA to avoid potential ROA maintenance overhead. If multiple prefixes are put into a ROA and then one of the prefixes in that ROA needs to change the origin AS, then the entire ROA needs to be deleted and recreated.

The optional field in the ROA called maxLength can be used to reduce the number of ROAs that need to be created. For example, if an ASN advertises four /24's that all fall under a /22, then one ROA for the /22 with a maxLength of /24 can be created instead of four separate ROAs for each /24. Most ISPs are either setting the maxLength value to the same as the prefix mask or not using this field. If the maxLength is set too large (e.g., /24 or /48) it opens the AS up to a potential forged-origin subprefix hijack (see [maxLength]).

ROAs are created by a CA. Information is fed into the CA by either an end user or an API, and the CA then adds the necessary digital signatures to the digital certificate. Once the CA has generated the ROA, a publication request is made to the publication server. If the publication server accepts the request, it will send back a repository response and generate the proper manifest, which will be retrieved by the validators that will connect to it. See [RFC 6486] for more information. After this connection, the repository will serve up the ROAs and manifest, CRL, and certificates via rsync and HTTPS to the validator clients. In HTTPS, the validator clients use the RPKI Repository Delta Protocol (RRDP) to retrieve the information hosted on the repository. See [RFC 8182] for more information.

If a prefix holder has a DDoS mitigation service provider that uses BGP route injection to divert traffic for scrubbing, that prefix holder will need to create ROAs to allow the DDoS mitigation service provider to originate that ASN's prefixes. This requirement applies to any situation in which another ASN will be validly originating the prefix holder's prefixes.

6.5 Monitor for Invalid Routes

After ROAs are generated and ROV is deployed, the prefix holder may need to develop some tools to monitor the output from ROV locally and remotely (such as RIPE's routing information service (RIS)), particularly the invalid and unknown routes. If an invalid or unknown route involves a new prefix being advertised, the prefix holder needs to generate a new ROA to make it valid.

7 DEPLOYMENT OF ROUTE ORIGIN VALIDATION (ROV)

Implementing ROV on a network involves five high-level steps.

1. Investigate ROV support on routing platforms where inter-provider BGP sessions are terminated. A reference for router support of ROV is available (https://rpki.readthedocs.io/en/latest/ops/router-support.html#doc-rpki-rtr), but operators should check with their own vendors.

   Running ROV on internal BGP sessions (including between autonomous systems run by the same organization) generally is not recommended because there are many cases in which more specific routes or routes with a private AS the origin are advertised over the internal BGP session.

2. Set up and operate a relying party system, which consists of RP software running on a server or, preferably, multiple servers. If multiple servers are deployed, they may run different RP software packages and may be placed in different geographic locations to achieve software diversity and location redundancy.

   The RP system needs to implement two functions, which can be in the same package/server or separate ones.

   - Collection and validation of ROAs produces a list of validated ROA payloads.
   - Feed the results to routers using the RPKI to Router (RTR) Protocol.

   Independent implementations of RP software are available (https://rpki.readthedocs.io/en/latest/ops/tools.html).

3. Design modified routing policies to properly include a "drop-invalid" posture. Many ISPs are adopting one of the following three approaches.

   - Implement a policy to tag invalids with a BGP community, then later change to reject after a few weeks of monitoring.
   - Slowly implement the rejection of invalids on routers over the course of a couple of months.
   - Run a simulation using traffic analysis, and deploy in one day.

   Depending on the structure of inbound policies, consider supporting a customer-triggered real-time black hole (RTBH) function.

4. Check internal routing tables for impacts from dropping invalids. For example, the RPKI Origin Validation Checker (https://github.com/job/rpki-ov-checker) can be used to check which prefixes with what origin autonomous systems are impacted.

5. Do not use ROV on internal sessions. For example, if private autonomous systems are used internally, they will be stripped when advertised externally; internally, the source will not match the ROAs.

8 MONITORING OF RPKI

There are several reasons to monitor both BGP announcements and RPKI itself. First, monitoring BGP announcements allows the ground truth of prefix ownership to be established, which is the first step to the successful deployment of ROAs. Second, such monitoring can also identify discrepancies between published ROAs and announced prefixes, facilitating the correction of potential human errors and misconfiguration. Third, monitoring of RPKI itself (e.g., ROV) may allow for the discovery of other issues that may go undetected.

There are two types of RPKI monitoring—external monitoring and internal monitoring.

8.1 External Monitoring

In external monitoring, BGP-related resources are monitored from outside the AS. More specifically, BGP announcements can be monitored from public route collecting systems, and ROAs can be monitored from public validators or the trust anchors/RIRs.

8.1.1 Monitoring of BGP Announcements

Public route collectors, such as RIPE's RIS (https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris), peer with many ASNs and collect BGP announcements from them to offer an external BGP viewpoint. For example, the prefixes announced by any ASN can be queried. Note that an operator using multiple ASNs that peer with each other can observe the prefixes announced by one ASN from the viewpoint of another.

The IRR Explorer tool (https://irrexplorer.nlnog.net/) reports possible issues with prefixes that an ASN announces in BGP for which there are stale or missing route objects in an IRR. The tool now also supports RPKI and lists all ROAs for those prefixes, which makes it a great tool to use if there are missing ROAs or if stale ROAs are being sent to an ASN, even if those prefixes are no longer controlled in house.

Other BGP monitoring tools include BGPAlerter, from NTT, and ROAST, under development by ISOC.

8.1.2 Monitoring of ROAs

Public validators such as the Cloudflare validator (https://rpki.cloudflare.com/?view=validator) allow one to query the public trust anchor lists (TALs) to view ROA validation state on a per-prefix or per-ASN level for any signed prefix. This action will validate a prefix to a ROA, even if the prefix is not announced; but it does not provide information on where the prefix actually is announced.

Trust anchor RIRs may provide an overview of prefixes allocated to an ASN. They work in the same way as public validators—by offering to query the TAL for ROAs and then comparing them to global peering announcements seen by route collectors. Misallocated PA space is also shown here.

8.2 Internal Monitoring

Aside from the obvious monitoring of configuration and machine health, it is also beneficial to internally monitor ROV states. Monitoring ROV on routers can highlight the routes customers or peers are sending that are being marked by ROV as invalid and dropped. This may be useful for the purposes of reporting (number of invalid routes received/dropped) or investigating misconfigurations. For example, if a customer is unable to reach an endpoint, checking if the route to the endpoint has been dropped by ROV can help resolve the problem more quickly.

ROV could be monitored by using the CLI, and other tools are also available. Job Snijders (https://github.com/job) developed an OV-checker script that can be used to monitor the validation state of received prefixes. OpenBMP, Artemis, and PMACCT can compare route tables by using BMP against an RTR server to identify RPKI invalid routes that are received and dropped. Telemetry may enable invalid prefix monitoring, but it must be configured on the network.

9 PEERING REQUIREMENTS

Because of the distributed natures of BGP and RPKI, the more networks that deploy RPKI, the more secure the BGP is. Therefore, when deploying RPKI in networks, consider encouraging or requiring neighbors to do the same. For example, consider adding the requirement of RPKI deployment into the peering agreements with neighbors.

Ensure good communication with neighbors regarding network maintenance, encourage them to actively cooperate to resolve security incidents and other operational problems, and, in principle, follow the practices outlined in BCP38 [RFC 2827] and BCP46 [RFC 3013]. Also consider asking business partners such as suppliers (e.g., cloud service provider) to also deploy RPKI to ensure that services are not disrupted by BGP hijacking.

10 SECURITY CONSIDERATIONS

RPKI is effective in mitigating prefix hijacking by verifying route announcements against cryptographically verified prefix and ASN pairs. However, there are other types of attacks against BGP that RPKI is not designed to mitigate, and RPKI itself introduces new risks that need to be mitigated.

RPKI is not designed to mitigate attacks involving the manipulation of the AS_PATH of a route. For example, an attacker can try to hijack an IP prefix by including the legitimate ASN as the first ASN (i.e., the origin AS) in the AS_PATH and the attacker's ASN as the second ASN. In this way, the malicious route will be considered valid based on ROA. With a shorter AS_PATH, the malicious route may be favored over other legitimate ones, resulting in prefix hijacking.

Further, if a ROA has a maxLength and the legitimately announced routes are less specific than allowed by the maxLength, a malicious AS can take advantage of the maxLength to announce more specific routes. By combing the more specific route with the manipulated AS_PATH, the malicious route can win the route selection against the legitimate routes. Although a more specific route has smaller address space than a less specific route, an attacker can announce multiple routes that are more specific to hijack the entire address space covered by a less specific route.

IETF is working on several solutions to mitigate manipulation of the AS_PATH, including BGPsec and ASPA. Because BGPsec requires BGP speakers to perform cryptographic operations, it is not considered practical; thus, it has not been deployed by operators. ASPA extends ROA to add neighboring ASNs to facilitate the validation of AS_PATH. Because ASPA piggybacks on existing ROAs and does not require BGP speakers to perform cryptographic operations, it is considered more promising.

RPKI is an infrastructure that is deployed outside of BGP. From a security perspective, RPKI introduces new attack surfaces that should not be ignored.

- First, the ROAs may contain incorrect ASN and prefix pairing [Gilad et al.], such as those due to human errors or incorrect data sources from which a ROA is created. Incorrect ROAs could lead to correct BGP routes being verified as invalid, resulting in legitimate traffic being dropped.
- Second, RPKI repositories storing ROAs could be tampered with to cause problems for ROV. Although ROAs themselves are digitally signed and cannot be tampered with, the repositories could be manipulated, such as by deleting some ROAs, to cause ROV to use incomplete ROAs. In this case, a prefix could be hijacked successfully if the corresponding ROA is made unavailable.
- Third, RPKI validator software may contain vulnerabilities (e.g., insufficient input validation) that can be exploited to attack BGP. Therefore, it is important for the developers of RPKI validators to follow the best practice for secure software development to minimize security vulnerabilities in validators' code bases.

Appendix I Acknowledgements

We wish to thank the following participants contributing directly to this document.

Contributor	Company Affiliation
Rich Compton	Charter Communications
Tony Tauber	Comcast
Mike Booth	Liberty Global
Mark Goodwin	Cox Communications
Tao Wan	CableLabs

We wish to thank Marilynn Court and Melanie Parker from CableLabs for their editing of this document and the following participants who reviewed this document.

Contributor	Company Affiliation
Tom Prevost	Armstrong
Shaun Feldbusch	Armstrong
Scott W Brendlinger	Armstrong
Brian Scriber	CableLabs
Matt Yahna	CableLabs
Rich Compton	Charter Communications
Jody Beck	Charter Communications
Daniel Schatte	Charter Communications
Stephen Smith	Charter Communications
Ryan Wilfling	Charter Communications
Tosin Ogunsanwo	Comcast
Courtney Smith	Comcast
Vaibhav Garg	Comcast
Jason Livingood	Comcast
Mazen Khaddam	Cox Communications
Amrish Patel	Cox Communications
Linzeng Zhang	Eastlink
Larry Brinton	Eastlink
Yasunori Motonaga	JCOM
Antoin Verschuren	Liberty Global
Kick Fronenbroek	Liberty Global
Steven Van Steen	Liberty Global
John Lubeck	Midco
Mike Bosma	Midco
Miles McCredie	Midco
Matt Tooley	NCTA
Colin Christie-White	Shaw Communications
Sabih Alsayed	Shaw Communications
Mariano D. Medina Lucena	Telcom Argentina
Philippe Couture	Videotron
Alejandro Garin	Videotron
Vincent Campeau	Videotron
Aaziz id Belhaj	Videotron
Sebastien Vigneault	Videotron

* * *