The RPKI Documentation

Alex Band and the RPKI Community

Jul 21, 2023

GENERAL

Welcome to the documentation of the Resource Public Key Infrastructure (RPKI), the community-driven technology based on open standards that is aimed at making Internet routing more secure. If you are new to this documentation, we recommend that you read the *introduction page* to get an overview of what this documentation has to offer.

Note: This documentation is an open source project maintained by the RPKI team at NLnet Labs, with contributions from the network operator community around the world. We always appreciate your feedback and improvements.

You can submit an issue or pull request on the GitHub repository, post a message on the RPKI mailing list or discuss RPKI on Discord.

The main documentation is organised into the following sections:

CHAPTER

ONE

INTRODUCTION

Welcome to the documentation of the Resource Public Key Infrastructure (RPKI). These pages offer a broad overview of the RPKI and how it can help make Internet routing using the Border Gateway Protocol (BGP) more secure. This way, you will learn how RPKI can benefit your organisation, as well as helping others to be more secure on the Internet.

1.1 About this Documentation

This documentation is continuously written, corrected and edited by the RPKI team at NLnet Labs. An initial version was written by Alex Band, Tim Bruijnzeels and Martin Hoffmann. Over time, additions from the network operator community, researchers and interested parties around the world were contributed. The documentation is edited via text files in the reStructuredText markup language and then compiled into a static website/offline document using the open source Sphinx and ReadTheDocs tools.

Note: You can contribute to the RPKI documentation by opening an issue or sending patches via pull requests on the GitHub source repository.

All the contents are under the permissive Creative Commons Attribution 3.0 (CC-BY 3.0) license, with attribution to “The RPKI team at NLnet Labs and the RPKI community”.

1.2 About Resource Public Key Infrastructure

RPKI allows holders of Internet number resources to make verifiable statements about how they intend to use their resources. To achieve this, it uses a public key infrastructure that creates a chain of resource certificates that follows the same structure as the way IP addresses and AS numbers are handed down.

RPKI is used to make Internet routing more secure. It is a community-driven system in which open source software developers, router vendors and all five Regional Internet Registries (RIRs) participate, i.e. ARIN, APNIC, AFRINIC, LACNIC and RIPE NCC.

Currently, RPKI is used to let the legitimate holder of a block of IP addresses make an authoritative statement about which AS is authorised to originate their prefix in the BGP. In turn, other network operators can download and validate these statements and make routing decisions based on them. This process is referred to as route origin validation (ROV). This provides a stepping stone to provide path validation in the future.

1.3 Organisation of this Documentation

This documentation is organised into three main sections:

- The *General* section contains this introduction as well as information about the licensing, authors, etc. It also contains the *FAQ* and the *Quick Help*.
- The *RPKI Technology* section explains the RPKI technology and standards in order for you to get a good sense of the requirements and moving parts. It will help you choose the right RPKI solution for your organisation, with regards to generating, publishing and using RPKI data.
- The *Operations* section is about various open source projects that are maintained to support RPKI, as well as router support and external resources.

CHAPTER

TWO

FAQ

2.1 RPKI Mechanism

2.1.1 What is RPKI and why was it developed?

The global routing system of the Internet consists of a number of functionally independent actors (Autonomous Systems) which use BGP (Border Gateway Protocol) to exchange routing information. The system is very dynamic and flexible by design. Connectivity and routing topologies are subject to change. Changes easily propagate globally within a few minutes. One weakness of this system is that these changes cannot be validated against information existing outside of the BGP protocol itself.

RPKI is a way to define data in an out-of-band system such that the information that are exchanged by BGP can be validated to be correct. The RPKI standards were developed by the IETF (Internet Engineering Task Force) to describe some of the resources of the Internet's routing and addressing scheme in a cryptographic system. These information are public, and anyone can get access to validate their integrity using cryptographic methods.

2.1.2 I thought we were all using the IRR to check route origin, why do we need RPKI now?

If you've been involved in default-free zone Internet engineering for any length of time, you're probably familiar with RPSL, a routing policy specification language originally defined in **RFC 2280** back in 1998. While RPSL has created considerable early enthusiasm and has seen some traction, the Internet was rapidly growing at the time, and the primary focus was on data availability rather than data trustworthiness. Everyone was busy opportunistically documenting the minimal policy that was necessary to "make things work" with the policy specification language parsing scripts of everyone else so that something would finally ping!

Over time, this has created an extensive repository of obsolete data of uncertain validity spread across dozens of route registries around the world. Additionally, the RPSL language and supporting tools have proven to be too complex to consistently transpose policy into router configuration language - resulting in most published RPSL data being neither sufficiently accurate and up to date for filtering purposes, nor sufficiently comprehensive or precise for being the golden master in router configuration.

RPKI aims to complement and expand upon this effort focusing primarily on trustworthiness, timeliness, and accuracy of data. RPKI ROAs are hierarchically delegated by RIRs based on strict criteria, and are cryptographically verifiable. This offers the Internet community an opportunity to build an up to date and accurate information of IP address origination data on the Internet.

2.1.3 Why are we investing in RPKI, isn't it easier to just fix the Internet Routing Registry (IRR) system?

The main weakness of the IRR is that it is not a globally deployed system and it lacks the authorisation model to make the system water tight. The result is that out of all the information on routing intent that is published, it is difficult to determine what is legitimate, authentic data and what isn't. RPKI solves these two problems, as you can be absolutely sure that an authoritative, cryptographically verifiable statement can be made by any legitimate IP resource holder in the world.

2.1.4 Is it true that BGP4 is just not up to the task any longer?

Unfortunately it's practically impossible to replace BGP right now. We should, however, work on fixing the broken parts and improving the situation.

2.1.5 As RPKI relies on X.509 PKI, isn't this the same problem with untrustworthy SSL/TLS Certificate Authorities all over again?

Instead of relying on a large number of CAs subject to variable auditing standards which come pre-installed in a browser or an operating system, RPKI relies on just five Trust Anchors, run by the Regional Internet Registries.

These are well established and openly governed organisations. Each operator that wishes to get an RPKI resource certificate already has a contractual relationship with one or more of the RIRs.

2.1.6 What is the value of RPKI based BGP Origin Validation without Path Validation?

While Path Validation is a desirable characteristic, the existing RPKI origin validation functionality addresses a large portion of the problem surface.

Existing operational and economic incentives ensure that the most important prefixes for each network are seen via the shortest AS path possible. One such example are network operators setting a higher local preference for prefixes learned via an Internet exchange or private peers ("peerlock"). This reduces the risk that an invalid route could win the BGP route selection process even if it originates from an impersonated but correct origin AS.

For transit providers, direct interconnections and short AS paths are a defining characteristic, positioning them ideally to act on RPKI data and accept only valid routes for redistribution.

Furthermore, operational experience suggests that the vast majority of route hijacks are unintentional rather than malicious, and are caused by 'fat-fingering', where an operator accidentally originates a prefix they are not the holder of. Origin Validation would mitigate many of these problems.

While a malicious party willing to intentionally impersonate the origin AS could still take advantage of the lack of Path Validation in some circumstances, widespread RPKI Origin Validation implementation would make such instances easier to pinpoint and address.

2.1.7 When comparing the ROA data set to the announcements my router sees, what are possible outcomes?

In short, routes can have the state Valid, Invalid, or NotFound (a.k.a. Unknown).

- Valid: The route announcement is covered by at least one ROA
- Invalid: The prefix is announced from an unauthorised AS or the announcement is more specific than is allowed by the maximum length set in a ROA that matches the prefix and AS
- NotFound: The prefix in this announcement is not covered (or only partially covered) by an existing ROA

To understand how more specifics, less specifics and partial overlaps are treated, please refer to section 2 of **RFC 6811**.

2.1.8 I've heard the term "route leak" and "route hijack". What's the difference?

A route leak is a propagation of one or more routing announcements that are beyond their intended scope. That is an announcement from an Autonomous System (AS) of a learned BGP route to another AS is in violation of the intended policies of the receiver, the sender, and/or one of the ASes along the preceding AS path.

A route hijack is the unauthorised origination of a route.

Note that in either case, the cause may be accidental or malicious and in either case, the result can be path detours, redirection, or denial of services. For more information, please refer to **RFC 7908**.

2.1.9 If a ROA is cryptographically invalid, will it make my route invalid?

An invalid ROA means that the object did not pass cryptographic validation and is therefore discarded. The statement about routing that was made within the ROA is simply not taken into consideration. An invalid route on the other hand, is the result of a valid ROA, specifically one that had the outcome that a prefix is announced from an unauthorised AS or the announcement is more specific than is allowed by the maximum length set in a ROA that matches the prefix and AS.

2.2 Operations and Impact

2.2.1 Will my router have a problem with all of this cryptographic validation?

No, routers do not do any cryptographic operations to perform Route Origin Validation. The signatures are checked by external software, called Relying Party software or RPKI Validator, which feeds the processed data to the router over a light-weight protocol. This architecture causes minimal overhead for routers.

2.2.2 Does RPKI reduce the BGP convergence speed of my routers?

No, filtering based on an RPKI validated cache has a negligible influence on convergence speed. RPKI validation happens in parallel with route learning (for new prefixes which aren't yet in cache), and those prefixes will be marked as valid, invalid, or notfound (and the correct policy applied) as the information becomes available.

2.2.3 Why do I need rsync on my system to use a validator?

In the original standards, rsync was defined as the main means of distribution of RPKI data. While it has served the system well in the early years, rsync has several downsides:

- When RPKI relying party software is used on a client system, it has a dependency on rsync. Different versions and different supported options, such as `--contimeout`, cause unpredictable results. Furthermore, calling rsync is inefficient. It's an additional process and the output can only be verified by scanning the disk.
- Scaling becomes more and more problematic as the global RPKI data set grows and more operators download and validate data, as with rsync the server in involved in processing the differences.

To overcome these limitations the RRDP protocol was developed and standardised in **RFC 8182**, which relies on HTTPS. RRDP was specifically designed for scaling and allows CDNs to participate in serving the RPKI data set globally, at scale. In addition, HTTPS is well supported in programming languages so development of relying party software becomes easier and more robust.

Currently, almost all RPKI publication points support RRDP. All RPKI Validator implementations have RRDP support as well, and prefer using it over rsync.

2.2.4 The five RIRs provide a Hosted RPKI system, so why would I want to run a Delegated RPKI system myself instead?

The RPKI system was designed to be a distributed system, allowing each organisation to run their own CA and publish the certificate and ROAs themselves. The hosted RIR systems are in place to offer a low entry barrier into the system, allowing operators to gain operational experience before deciding if they want to run their own CA.

For many operators, the hosted system will be good enough, also in the long term. However, organisations who for example don't want to be dependent on a web interface for management, who manage address space across multiple RIR regions, or have BGP automation in place that they would like to integrate with ROA management, can all choose to run a CA on their own systems.

2.2.5 Should I run a validator myself, when I can use an external data source I found on the Internet?

The value of signing the authoritative statements about routing intent by the resource holder comes from being able to validate that the data is authentic and has not been tampered with in any way.

When you outsource the validation to a third party, you lose the certainty of data accuracy and authenticity. Conceptually, this is similar to DNSSEC validation, which is best done by a local trusted resolver.

Section 3 of **RFC 7115** has an extensive section on this specific topic.

2.2.6 How often should I fetch new data from the RPKI repositories?

According to section 3 of **RFC 7115** you should fetch new data at least every 4 to 6 hours. At the moment, the publication of new ROAs in the largest repositories takes about 10-15 minutes. This means fetching every 15-30 minutes is reasonable, without putting unnecessary load on the system.

2.2.7 What if the RPKI system becomes unavailable or some other catastrophe occurs, will my (signed) prefixes become unreachable to others? Will other prefixes my routers learned over BGP become unreachable for me?

RPKI provides a positive statement on routing intent. If all RPKI validator instances become unavailable and all certificates and ROAs expire, the validity state of all routes will fall back to NotFound, as if RPKI were never used. Routes with this state should be accepted according to section 5 of **RFC 7115**, as this state will unfortunately be true for the majority of routes.

2.2.8 What if the Validator I use crashes and my router stops getting a feed. What will happen to the prefixes I learn over BGP?

All routers that support Route Origin Validation allow you to specify multiple Validators for redundancy. It is recommended that you run multiple instances, preferably from independent publishers and on separate subnets. This way you rely on multiple caches.

In case of a complete failure, all routes will fall back to the NotFound state, as if Origin Validation were never used.

2.2.9 I don't want to rely on the RPKI data set in all cases, but I want to have my own preferences for some routes. What can I do?

You can always apply your own, local overrides on specific prefixes/announcements and override the RPKI data you fetch from the repositories. Specifying overrides is in fact standardised in **RFC 8416**, "Simplified Local Internet Number Resource Management with the RPKI (SLURM)".

2.2.10 Is there any point in signing my routes with ROAs if I don't validate and filter myself?

Yes, signing your routes is always a good idea. Even if you don't validate yourself someone else will, or in worst case someone else might try to hijack your prefix. Imagine what could happen if you haven't signed your prefixes…

2.3 Miscellaneous

2.3.1 Why isn't the ARIN RPKI TAL like other public key files?

Unlike the other RIRs, which distribute their TAL publicly, ARIN has a policy requiring users to explicitly agree to terms and conditions concerning its TAL. Note that this policy is not without controversy as discussed here and here on the NANOG list.

Job Snijders made a video explaining his perspective on the ARIN TAL. Christopher Yoo and David Wishnick authored a paper titled Lowering Legal Barriers to RPKI Adoption.

Ben Cox performed various RPKI measurements and concluded that the ARIN TAL is used far less than TALs from their RIR counter parts. This has led to a situation where ROAs created under the ARIN TAL offer less protection against BGP incidents than other RIRs. State of RPKI: Q4 2018.

2.3.2 What is the global adoption and data quality of RPKI like?

There are several initiatives that measure the adoption and data quality of RPKI:

- RPKI Analytics, by NLnet Labs
- Global certificate and ROA statistics, by RIPE NCC
- Cirrus Certificate Transparency Log, by Cloudflare
- The RPKI Observatory, by nusenu
- RPKI Deployment Monitor, by NIST

2.3.3 I want to use the RPKI services from a specific RIR that I'm not currently a member of. Can I transfer my resources?

The RPKI services that each RIR offers differ in conditions, terms of service, availability and usability. Most RIRs have a transfer policy that allow their members to transfer their resources from one RIR region to another. Organisations may wish to do this so that they bring all resources under one entity, simplifying management. Others may do this because they are are looking for a specific set of terms with regards to the holdership of their resources. Please check with your RIR for the possibilities and conditions for resource transfers.

2.3.4 Will RPKI be used as a censorship mechanism allowing governments to make arbitrary prefixes unroutable on a whim?

Unlikely. In order to suppress a prefix, it would be necessary to both revoke the existing ROA (if one is present) and publish a conflicting ROA with a different origin.

These characteristics make using RPKI as a mechanism for censorship a rather convoluted and uncertain way of achieving this goal, and has broad visibility (as the conflicting ROA, as well as the Regional Internet Registry under which it was issued, will be immediately accessible to everyone). A government would be much better off walking into the data center and confiscate your equipment.

2.3.5 What are the long-term plans for RPKI?

With RPKI Route Origin Validation being deployed in more and more places, there are several efforts to build upon this to offer out-of-band Path Validation. Autonomous System Provider Authorisation (ASPA) currently has the most traction in the IETF, defined in these drafts: draft-azimov-sidrops-aspa-profile and draft-azimov-sidrops-aspa-verification.

CHAPTER

THREE

QUICK HELP

If you're reading this page, chances are you find yourself in a situation where you've been told by someone that your RPKI ROAs make your routes invalid and you don't know what that means. The aim of the content on this page is to point you in the right direction and provide further resources that can be of assistance. This page is not meant for experts, and many technicalities will be glossed over in order to be able to provide easy to understand answers for all knowledge levels.

3.1 What is RPKI or ROA?

RPKI stands for Resource Public Key Infrastructure, ROA stands for Route Origin Authorisation.

3.2 What do they do?

They provide a method for the originator of a route to assert they are the correct originator and that other originators are not valid.

3.3 How does it work?

The "root" assigner of all IP space (v4+v6) is IANA. They have delegated this space to one of the RIRs (ARIN, RIPE NCC, APNIC, LACNIC, and AFRINIC). In turn, those RIRs assign the space to other entities. Each RIR has a portal where the owner of the space can assert the origination ASN, which then generates a ROA for that particular combination of route and origination ASN. This ROA is then published out by the RIR so that anyone can view them.

3.4 What is in a ROA?

A ROA is a signed statement that consists of a prefix, a maximum prefix length, and originating ASN.

3.5 What happens next?

Any operator is free to get that list of ROAs from the RIRs and use that to tell their routers to take action based on the ROA. A particular announcement will generally have one of three states:

NotFound (a.k.a. Unknown)
This is the default state if no ROA has been made for the announcement. It is expected that all operators will allow these routes to be installed in their routers.

Valid
This is the state if the ROA and route announcement matches. It is expected that all operators will allow these routes to be installed in their routers. It is possible they may up-preference these routes.

Invalid
This is the state if the ROA and route announcement are different. They either differ in originating ASN or is more specific than is allowed by the maximum prefix length that is set in the ROA. If an operator is using RPKI in a strict fashion, odds are good that this announcement will not be installed into their routers.

3.6 What can I do about my route having an Invalid state?

The only entity that can make any changes to the ROA is the RIR-listed owner of the IP space. Most likely the owner of the IP space has created their ROAs in the Hosted RPKI interface of the RIR, which is part of their respective member portals:

- AFRINIC: https://my.afrinic.net
- APNIC: https://myapnic.net
- ARIN: https://account.arin.net
- LACNIC: https://milacnic.lacnic.net
- RIPE NCC: https://my.ripe.net

It is important to note that initially, for there to be an RPKI Invalid route, someone must have already entered into one of the above portals and made a ROA for the IP space in question. There is no way for it to have to been done by itself. In other words, there must already be an account at the RIR that is linked to the owner of the IP space.

Note: Perhaps someone told you that your routes are not yet covered under a RPKI ROA (NotFound). The pointers in this section are equally applicable to the case where RPKI is completely new to you.

CHAPTER FOUR

INTRODUCTION

Resource Public Key Infrastructure (RPKI) revolves around the right to use Internet number resources, such as IP addresses and autonomous system (AS) numbers.

In this PKI, the legitimate holder of a block of IP addresses or AS numbers can obtain a resource certificate. Using the certificate, they can make authoritative, signed statements about the resources listed on it. To understand the structure of RPKI and its usage, we must first look at how Internet number resources are allocated globally.

4.1 Internet Number Resource Allocation

Before being formalised within an organisation, the allocation of Internet number resources, such as IP addresses and AS numbers, had been the responsibility of Jon Postel. At the time, he worked at the Information Sciences Institute (ISI) of the University of Southern California (USC). He performed the role of Internet Assigned Numbers Authority (IANA), which is presently a function of the Internet Corporation for Assigned Names and Numbers (ICANN).

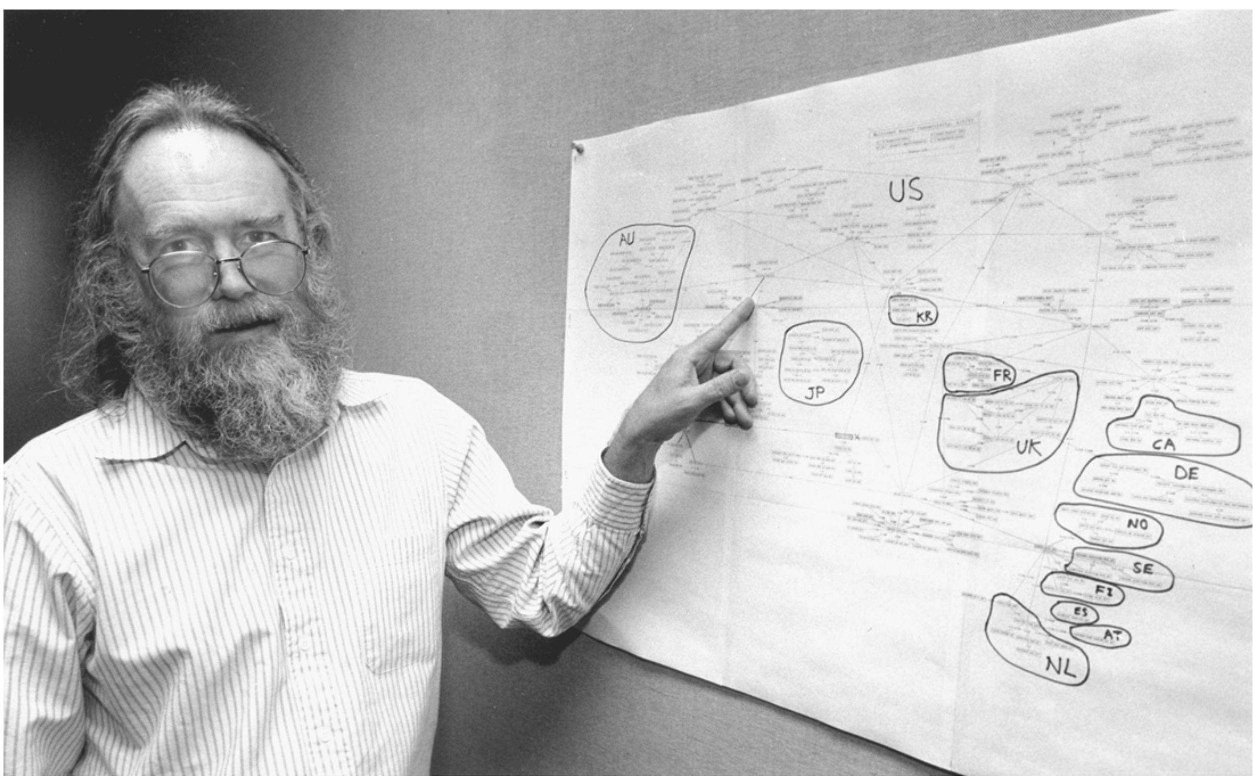

Fig. 4.1: Jon Postel in 1994, with a map of Internet top-level domains

Initially, the IANA function was performed globally, but as the work volume grew due to the expansion of the Internet, Regional Internet Registries (RIRs) were established over the years to take on this responsibility on a regional level. Until the available pool of IPv4 depleted in 2011, this meant that periodically, a large block of IPv4 address space was allocated from IANA to one of the RIRs. In turn, the RIRs would allocate smaller blocks to their member organisations, and so on. IPv6 address blocks and AS numbers are allocated in the same way.

Today, there are five RIRs responsible for the allocation and registration of Internet number resources within a particular region of the world:

- The African Network Information Center (AFRINIC) serves Africa
- The American Registry for Internet Numbers (ARIN) serves Antarctica, Canada, parts of the Caribbean, and the United States
- The Asia-Pacific Network Information Centre (APNIC) serves East Asia, Oceania, South Asia, and Southeast Asia
- The Latin America and Caribbean Network Information Centre (LACNIC) serves most of the Caribbean and all of Latin America
- The Réseaux IP Européens Network Coordination Centre (RIPE NCC) serves Europe, the Middle East, Russia, and parts of Central Asia

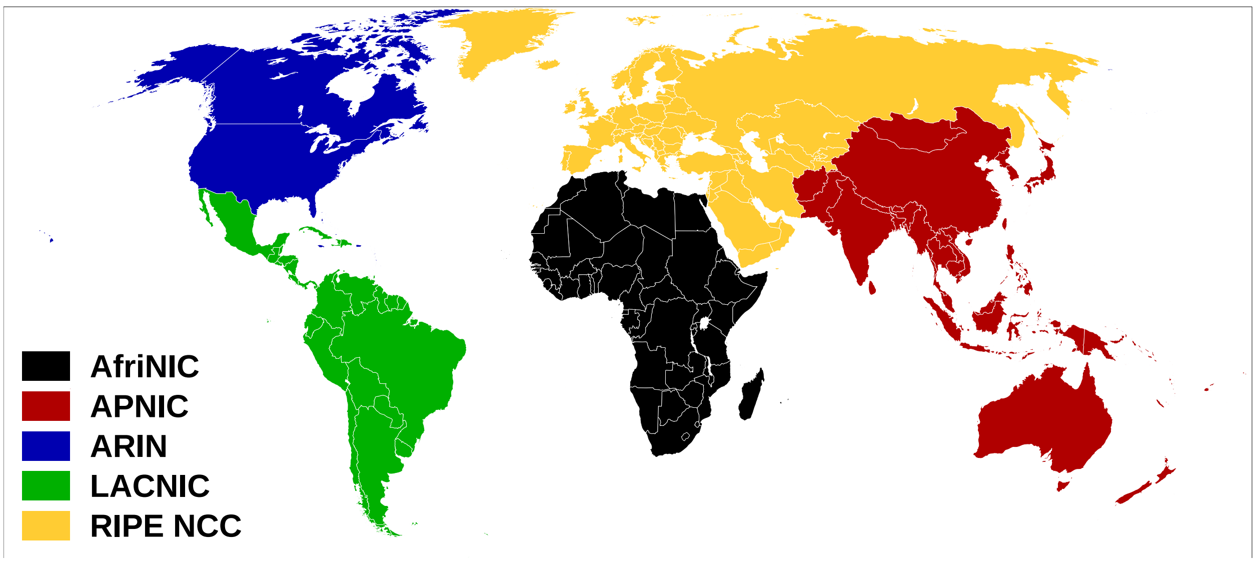

Fig. 4.2: The service regions of the five Regional Internet Registries

In the APNIC and LACNIC regions, Internet number resources are in some cases allocated to National Internet Registries (NIRs), such as NIC.br in Brazil and JPNIC in Japan. NIRs allocate address space to its members or constituents, which are generally organised at a national level. In the rest of world, the RIRs allocate directly to their member organisations, typically referred to as Local Internet Registries (LIRs). Most LIRs are Internet service providers, enterprises, or academic institutions. LIRs either use the allocated IP address blocks themselves, or assign them to End User organisations.

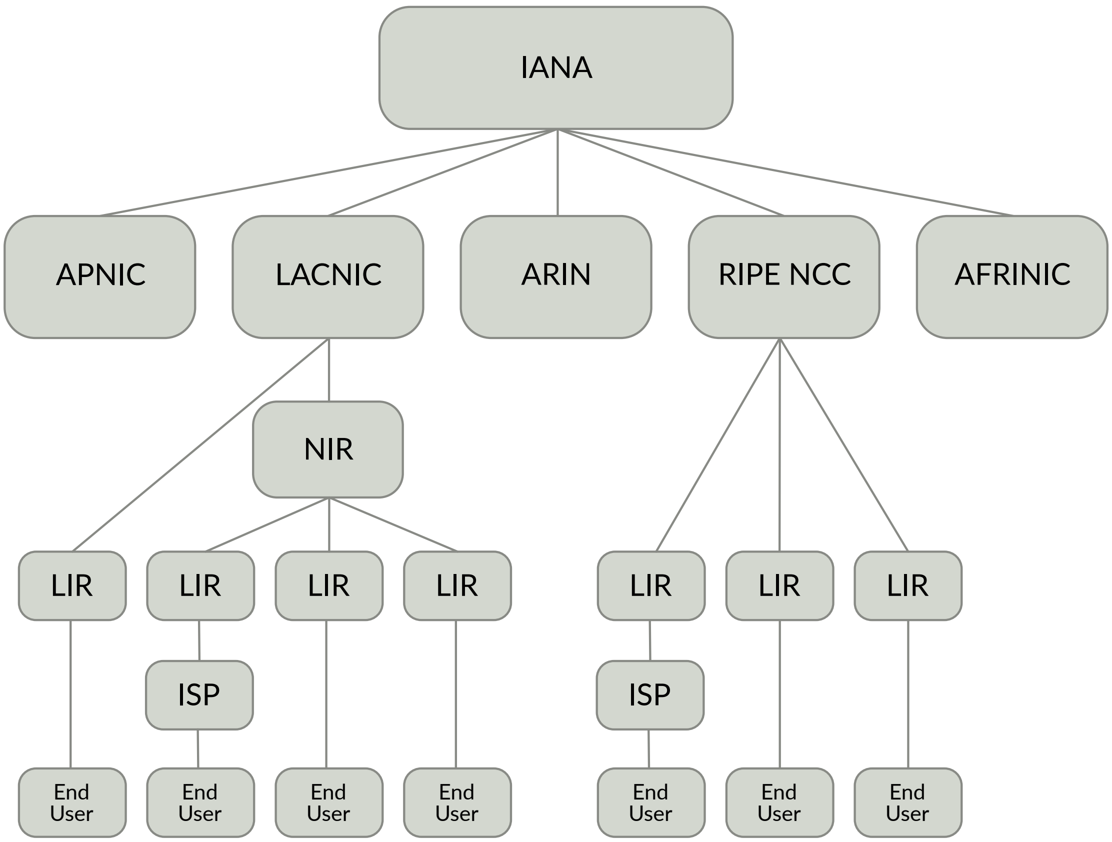

Fig. 4.3: Internet number resource allocation hierarchy

4.2 Mapping the Resource Allocation Hierarchy into the RPKI

As illustrated, the IANA has the authoritative registration of IPv4, IPv6 and AS number resources that are allocated to the five RIRs. Each RIR registers authoritative information on the allocations to NIRs and LIRs, and lastly, LIRs record to which End User organisation they assigned resources.

In RPKI, resource certificates attest to the allocation by the issuer of IP addresses or AS numbers to the subject. As a result, the certificate hierarchy in RPKI follows the same structure as the Internet number resource allocation hierarchy, with the exception of the IANA level. Instead, the five RIRs each run a root CA with a trust anchor from which a chain of trust for the resources they each manage is derived.

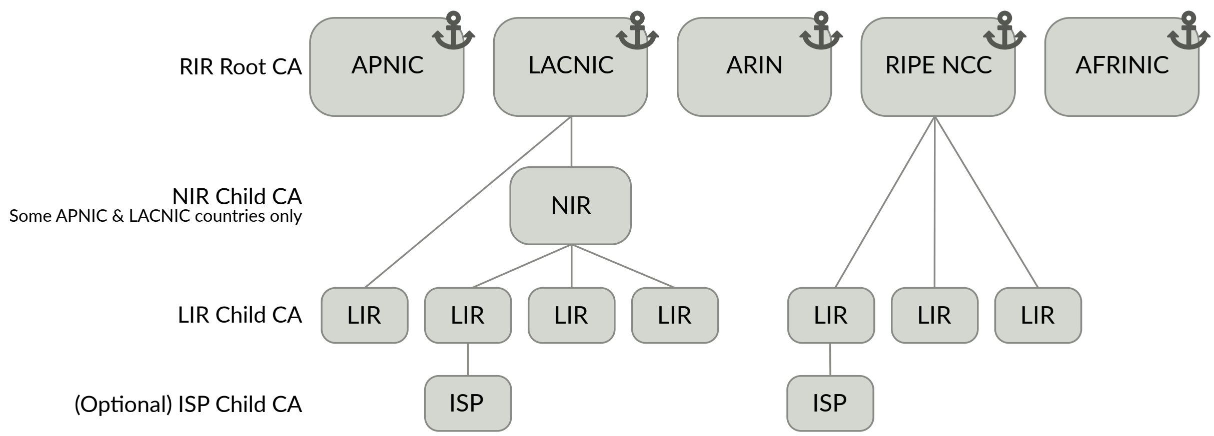

Fig. 4.4: The chain of trust in RPKI, starting at the five RIRs

The IANA does not operate a single root certificate authority (CA). While this was originally a recommendation from the Internet Architecture Board (IAB) to eliminate the possibility of resource conflicts in the system, they reconsidered after operational experience in deployment had caused the RIRs to conclude that the RPKI system would be less brittle using multiple overlapping trust anchors.

4.3 X.509 PKI Considerations

The digital certificates used in RPKI are based on X.509, standardised in **RFC 5280**, along with extensions for IP addresses and AS identifiers described in **RFC 3779**. Because RPKI is used in the routing security context, a common misconception is that this is the *Routing* PKI. However, certificates in this PKI are called *resource* certificates and conform to the certificate profile described in **RFC 6487**.

Note: X.509 certificates are typically used for authenticating either an individual or, for example, a website. **In RPKI, certificates do not include identity information**, as their only purpose is to transfer the right to use Internet number resources.

In addition to RPKI not having any identity information, there is another important difference with commonly used X.509 PKIs, such as SSL/TLS. Instead of having to rely on a vast number of root certificate authorities which come pre-installed in a browser or an operating system, RPKI relies on just five trust anchors, run by the RIRs. These are well established, openly governed, not-for-profit organisations. Each organisation that wishes to get an RPKI resource certificate already has a contractual relationship with one or more of the RIRs.

In conclusion, RPKI provides a mechanism to make strong, testable attestations about Internet number resources. In the next sections, we will look at how this can be used to make Internet routing more secure.

CHAPTER

FIVE

INTERNET ROUTING

To understand how RPKI is used to make Internet routing more secure, we must first look at how routing works, what the weaknesses are and which elements RPKI can currently help protect against.

The global routing system of the Internet consists of a number of functionally independent actors called autonomous systems (AS), which use the Border Gateway Protocol (BGP) to exchange routing information.

An autonomous system is a set of Internet routable IP prefixes belonging to a network or a collection of networks that are all managed and supervised by a single entity or organisation. An AS utilises a common routing policy controlled by the entity and is identified by a globally unique 16 or 32-bit number. The AS number (ASN) is assigned by one of the five Regional Internet Registries (RIRs), just like IP address blocks.

The Border Gateway Protocol manages the routed peerings, prefix advertisement and routing of packets between different autonomous systems across the Internet. BGP uses the ASN to uniquely identify each system. In short, BGP is the routing protocol for AS paths across the Internet. The system is very dynamic and flexible by design. Connectivity and routing topologies are subject to change, which easily propagate globally within a few minutes.

Fundamentally, BGP is based on mutual trust between networks. When a network operator configures the routers in their AS, they specify which IP prefixes to originate and announce to their peers. There is no authentication or authorisation embedded within BGP. In principle, an operator can define any ASN as the origin and announce any prefix, also one they are not the holder of.

5.1 BGP Best Path Selection

BGP routing information includes the complete route to each destination. BGP uses the routing information to maintain a database of network reachability information, which it exchanges with other networks. For each prefix in the routing table, BGP continuously and dynamically makes decisions about the best path to reach a particular destination. After the best path is selected, the route is installed in the routing table.

Though there are many factors at play, two of them are the most important to keep in mind throughout the next sections: the preference for shortest path and most specific IP prefix.

5.1.1 Preference for Shortest Path

Out of all the possible routes that a router has in its Routing Information Base (RIB), BGP will always prefer the shortest path to its destination, minimising the amount of hops. When two matching prefixes are announced from two different networks on the Internet, BGP will route traffic to the destination that is topologically closest. This is an important feature of BGP, but when configuration errors occur, it can also be the cause of reachability problems.

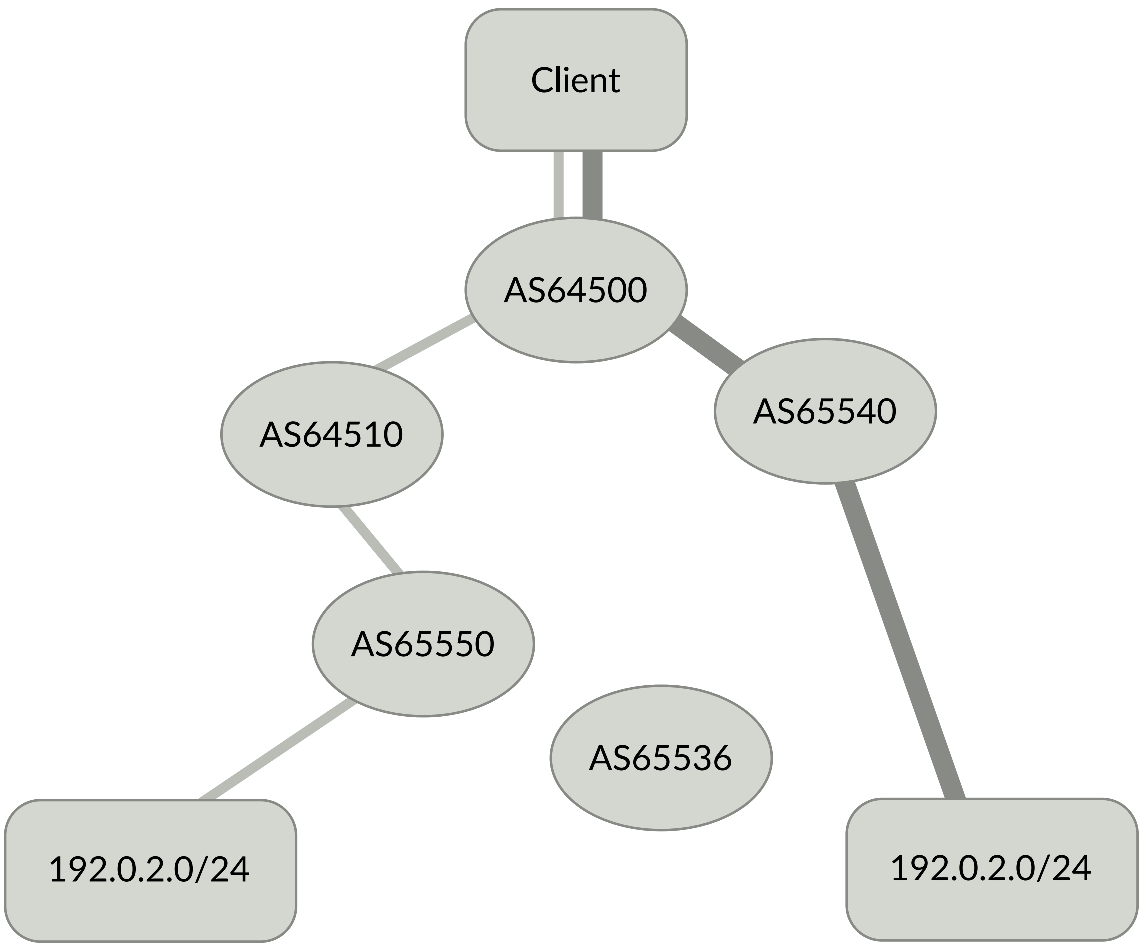

Fig. 5.1: When the announcement of a prefix is an exact match, the shortest path wins

5.1.2 Preference for Most Specific Prefix

Regardless any local preference, path length or any other attributes, when building the forwarding table, the router will always select most specific IP prefix available. This behaviour is important, but creates the possibility for almost any network to attract someone else's traffic by announcing an overlapping more specific.

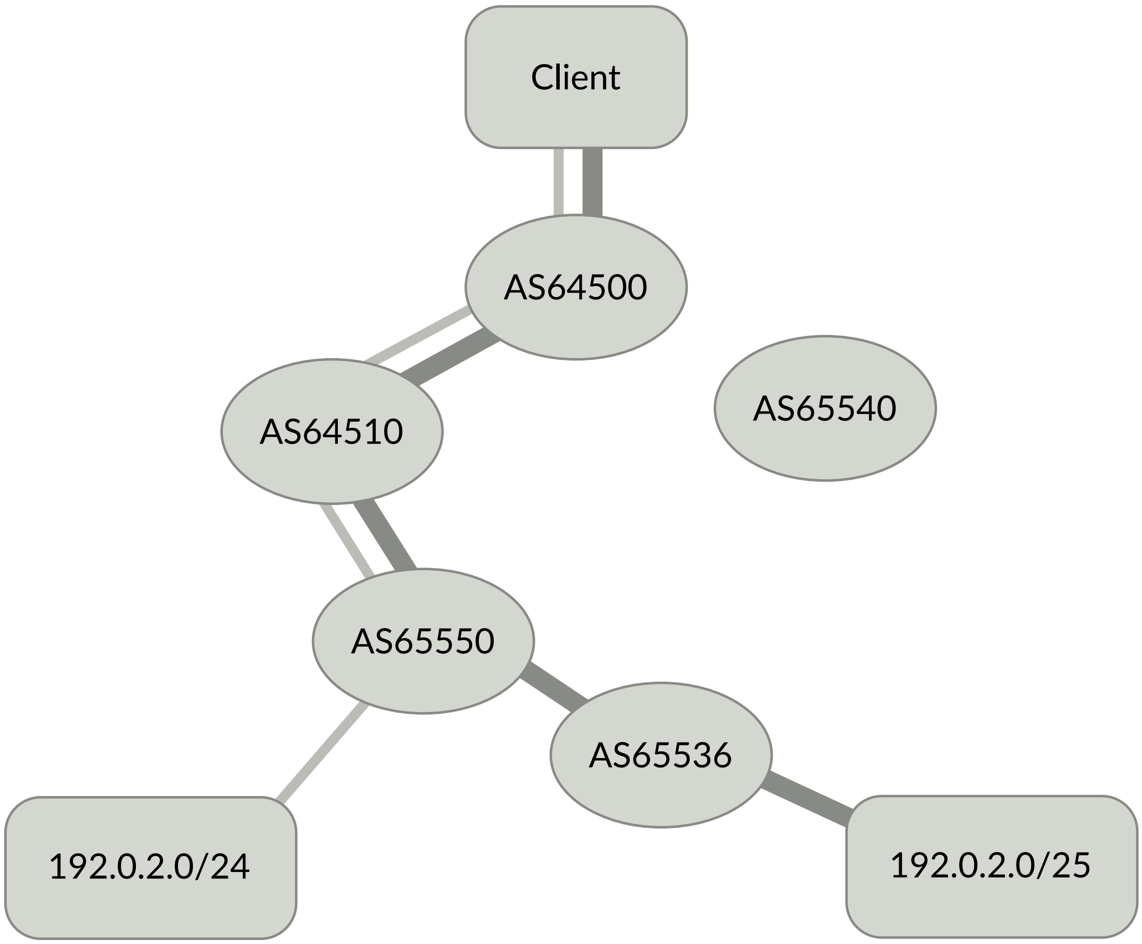

Fig. 5.2: Regardless of the path length, the announcement of a more specific prefix always wins

With this in mind, there are several problems that can arise as a result of this behaviour.

5.2 Routing Errors

Routing errors on the Internet can be classified as route leaks or route hijacks. **RFC 7908** provides a working definition of a BGP route leak:

> A route leak is the propagation of routing announcement(s) beyond their intended scope. That is, an announcement from an Autonomous System (AS) of a learned BGP route to another AS is in violation of the intended policies of the receiver, the sender, and/or one of the ASes along the preceding AS path. The intended scope is usually defined by a set of local redistribution/filtering policies distributed among the ASes involved. Often, these intended policies are defined in terms of the pair-wise peering business relationship between autonomous systems.

A route hijack, also called prefix hijack, or IP hijack, is the unauthorised origination of a route.

Note: Route leaks and hijacks can be accidental or malicious, but most often arise from **accidental misconfigurations**. The result can be redirection of traffic through an unintended path. This may enable eavesdropping or traffic analysis and may, in some cases, result in a denial of service or black hole.

Routing incidents occur every day. While several decades ago outages and redirections were often accidental, in recent years they have become more malicious in nature. Some notable events were the AS 7007 incident in 1997, Pakistan's attempt to block YouTube access within their country, which resulted in taking down YouTube entirely in 2008, and lastly, the almost 1,300 addresses for Amazon Route 53 that got rerouted for two hours in order to steal cryptocurrency, in 2018.

5.3 Mitigation of Routing Errors

One weakness of BGP is that routing errors cannot be easily be deduced from information within the protocol itself. For this reason, network operators have to carefully gauge what the intended routing policy of their peers is. As a result, it is imperative that networks employ filters to only accept legitimate traffic and drop everything else.

There are several well known methods to achieve this. Certain backbone and private peers require a valid Letter of Agency (LOA) to be completed prior to allowing the announcement or re-announcement of IP address blocks. A more widely accepted method is the use of Internet Routing Registry (IRR) databases, where operators can publish their routing policy. Both methods allow other networks to set up filters accordingly.

5.4 The Internet Routing Registry

The Internet Routing Registry (IRR) is a distributed set of databases allowing network operators to describe and query for routing intent. The IRR is used as a verification mechanism of route origination and is widely, though not universally, deployed to prevent accidental or intentional routing disturbances.

The notation used in the IRR is the Routing Policy Specification Language (RPSL), which was originally defined in **RFC 2280** in 1998. RPSL is a very expressive language, allowing for an extremely detailed description of routing policy. While IRR usage had created considerable early enthusiasm and has seen quite some traction, the Internet was rapidly growing at the time. This meant that the primary focus was on data availability rather than data trustworthiness.

In later years, it was considered a good practice to extensively document how incoming and outgoing traffic was treated by the network, but nowadays the most prevalent usage is to publish and query for *route* objects, describing from which ASN a prefix is intended to be originated:

```
route:          192.0.2.0/24
descr:          Examplenet announcement of 192.0.2.0/24
country:        NL
origin:         AS65536
mnt-by:         EXAMPLENET-MNT
mnt-routes:     EXAMPLENET-MNT
last-modified:  2018-08-30T07:50:19Z
source:         RIPE
```

As explained earlier, only the Regional Internet Registries have authoritative information on the legitimate holder of an Internet number resource. This means that the entries in their IRR databases are authenticated, but they are not in any of the other routing registries. Over time, this has created an expansive repository of obsolete data of uncertain validity, spread across dozens of routing registries around the world.

Additionally, the RPSL language and supporting tools have proven to be too complex to consistently transpose policy into router configuration language. This resulted in most published RPSL data being neither sufficiently accurate and up to date for filtering purposes, nor sufficiently comprehensive or precise for being the golden master in router configuration.

In conclusion, the main weakness of the IRR is that it is not a globally deployed system and it lacks the authorisation model to make the system water tight. The result is that out of all the information on routing intent that is published, it is difficult to determine what is legitimate, authentic data and what isn't.

RPKI solves these problems, as you can be absolutely sure that an authoritative, cryptographically verifiable statement can be made by any legitimate IP resource holder in the world. In the next sections we will look at how this is achieved.

CHAPTER

SIX

SECURING BGP

Now that we've looked at how the RPKI structure is built and understand the basics of Internet routing, we can look at how RPKI can be used to make BGP more secure.

RPKI provides a set of building blocks allowing for various levels of protection of the routing system. The initial goal is to provide route origin validation, offering a stepping stone to providing path validation in the future. Both origin validation and path validation are documented IETF standards. In addition, there are drafts describing autonomous system provider authorisation, aimed at providing a more lightweight, incremental approach to path validation.

6.1 Route Origin Validation

With route origin validation (ROV), the RPKI system tries to closely mimic what *route* objects in the IRR intend to do, but then in a more trustworthy manner. It also adds a couple of useful features.

Origin validation is currently the only functionality that is operationally used. The five RIRs provide functionality for it, there is open source software available for creation, publication and use of data, and all major router vendors have implemented ROV in their platforms. Various router software implementations offer support for it, as well.

6.1.1 Route Origin Authorisations

Using the RPKI system, the legitimate holder of a block of IP addresses can use their resource certificate to make an authoritative, signed statement about which autonomous system is authorised to originate their prefix in BGP. These statements are called Route Origin Authorisations (ROAs).

The creation of a ROA is solely tied to the IP address space that is listed on the certificate and not to the AS numbers. This means the holder of the certificate can authorise any AS to originate their prefix, not just their own autonomous systems.

Maximum Prefix Length

In addition to the origin AS and the prefix, the ROA contains a maximum length (maxLength) value. This is an attribute that a *route* object in RPSL doesn't have. Described in **RFC 6482**, the maxLength specifies the maximum length of the IP address prefix that the AS is authorised to advertise. This gives the holder of the prefix control over the level of deaggregation an AS is allowed to do.

For example, if a ROA authorises a certain AS to originate 192.0.1.0/24 and the maxLength is set to /25, the AS can originate a single /24 or two adjacent /25 blocks. Any more specific announcement is unauthorised by the ROA. Using this example, the shorthand notation for prefix and maxLength you will often encounter is `192.0.1.0/24-25`.

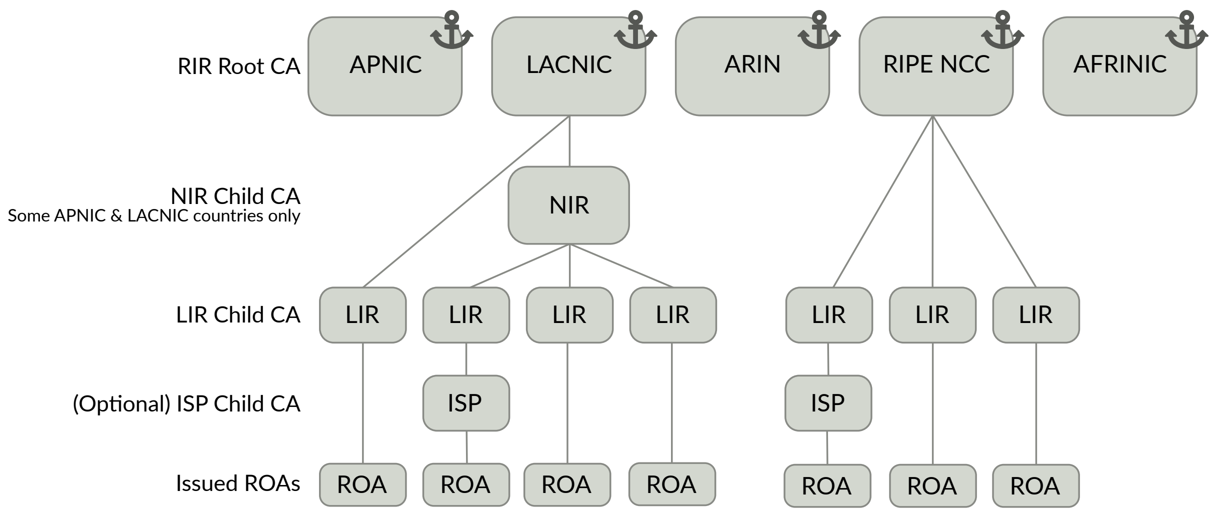

Fig. 6.1: Each CA can issue Route Origin Authorisations

> **Warning:** According to **RFC 7115**, operators should be conservative in use of maxLength in ROAs. For example, if a prefix will have only a few sub-prefixes announced, multiple ROAs for the specific announcements should be used as opposed to one ROA with a long maxLength.
>
> **Liberal usage of maxLength opens up the network to a forged origin attack. ROAs should be as precise as possible, meaning they should match prefixes as announced in BGP.**

In a forged origin attack, a malicious actor spoofs the AS number of another network. With a minimal ROA length, the attack does not work for sub-prefixes that are not covered by overly long maxLength. For example, if, instead of 10.0.0.0/16-24, one issues 10.0.0.0/16 and 10.0.42.0/24, a forged origin attack cannot succeed against 10.0.66.0/24. They must attack the whole /16, which is more likely to be noticed because of its size.

6.1.2 Route Announcement Validity

When a network operator creates a ROA for a certain combination of origin AS and prefix, this will have an effect on the RPKI validity of one or more route announcements. Once a ROA is validated, the resulting object contains an IP prefix, a maximum length, and an origin AS number. This object is referred to as validated ROA payload (VRP).

When comparing VRPs to route announcements seen in BGP, **RFC 6811** describes their possible statuses, which are:

Valid
: The route announcement is covered by at least one VRP. The term *covered* means that the prefix in the route announcement is equal, or more specific than the prefix in the VRP.

Invalid
: The prefix is announced from an unauthorised AS, or the announcement is more specific than is allowed by the maxLength set in a VRP that matches the prefix and AS.

NotFound
: The prefix in this announcement is not, or only partially covered by a VRP.

Anyone can download and validate the published certificates and ROAs and make routing decisions based on these three outcomes. In the *Using RPKI Data* section, we'll cover how this works in practice.

6.2 Path Validation

Currently, RPKI only provides origin validation. While BGPsec path validation is a desirable characteristic and standardised in **RFC 8205**, real-world deployment may prove limited for the foreseeable future. However, RPKI origin validation functionality addresses a large portion of the problem surface.

For many networks, the most important prefixes can be found one AS hop away (coming from a specific peer, for example), and this is the case for large portions of the Internet from the perspective of a transit provider - entities which are ideally situated to act on RPKI data and accept only valid routes for redistribution.

Furthermore, the vast majority of route hijacks are unintentional, and are caused by 'fat-fingering', where an operator accidentally originates a prefix they are not the holder of.

Origin validation would mitigate most of these problems, offering immediate value of the system. While a malicious party could still take advantage of the lack of path validation, widespread RPKI implementation would make such instances easier to pinpoint and address.

With origin validation being deployed in more and more places, there are several efforts to build upon this to offer out-of-band path validation. Autonomous system provider authorisation (ASPA) currently has the most traction in the IETF, and is described in these drafts: draft-azimov-sidrops-aspa-profile and draft-azimov-sidrops-aspa-verification.

CHAPTER

SEVEN

IMPLEMENTATION MODELS

RPKI is designed to allow every resource holder to generate and publish cryptographic material on their own systems. This is commonly referred to as delegated RPKI. To offer a turn-key solution, each RIR also offers a hosted RPKI system in their member portals. Both models have their own advantages, based on the specific requirements of the organisation using the system.

No matter what implementation model you choose, it always a good idea to publish ROAs for your BGP announcements. Even when you are still evaluating how to deploy RPKI within your organisation, the benefits are immediate. Others can already filter based on what you publish, offering protection for you and other Internet users. For example, in case someone inadvertently announces your address space from their AS, it will be flagged as Invalid and dropped by everyone who has deployed route origin validation.

Important: Once you start authorising announcements with RPKI, it is imperative that ROAs are created for all route origins from the prefixes you hold, including more specifics announced by other business units or customers. In addition, RPKI should become a standard part of operations, ensuring staff is trained and ROAs are continually monitored and maintained.

7.1 Hosted RPKI

In 2008, when the five RIRs committed to start offering RPKI services, it was clear that there would be an early adopters phase for a considerable amount of time. Given the past experiences with IPv6 and DNSSEC uptake, the RIRs decided to offer a hosted RPKI solution to lower the entry barrier into the technology. This way, organisations could easily get operational experience with the technology, without having to manage a certificate authority themselves.

Hosted RPKI offers a fair balance between ease-of-use, maintenance and flexibility. It allows users to log into their RIR member portal and request a resource certificate, which is securely hosted on the servers of the RIR. All cryptographic operations, such as key roll overs, are automated. The certificates and ROA are published in repositories hosted by the RIR. In short, there is nothing that the user has to manage, apart from creating and maintaining ROAs.

The functionality and user interfaces of the hosted RPKI implementations vary greatly across the five RIRs. Despite these variations, if you are an organisation with a single ASN and a handful of statically announced IP address blocks that are not delegated to customers, hosted RPKI is sufficient for most use cases.

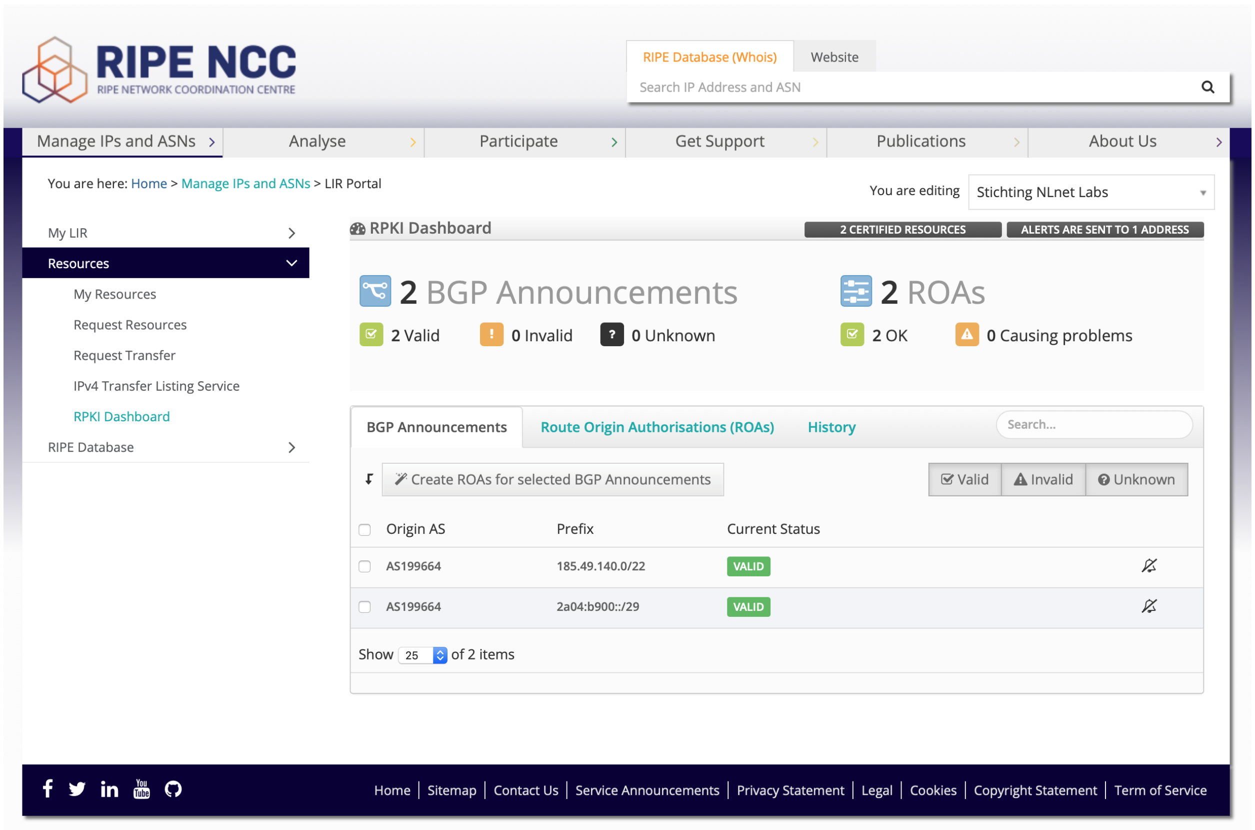

Fig. 7.1: Example of the Hosted RPKI interface of the RIPE NCC

7.1.1 Functional differences across RIRs

This section provides an overview of the functionality each RIR provides to help users manage RPKI, which is summarised in the table below.

First, the table indicates if the RPKI system supports setting up delegated RPKI, so users can run their own certificate authority if they want. An RIR may also offer a publication server for users running delegated RPKI. When using the hosted RPKI system, there is an overview if multiple users can be authorised to manage ROAs, and whether they can authenticate using two-factors.

To make management of ROAs easier, some systems provide a list of all announcements with certified address space that are seen by BGP route collectors, such as the RIPE Routing Information Service (RIS). ROAs have an explicit start and end validity date, but in some cases it is possible to automatically renew the ROAs, so that they are valid for as long as there is an entry in the web interface. In addition, it may be possible to synchronise the management of *"route"* objects in the IRR with the ROAs that are created. An application programming interface (API) may be provided to make batch processing easier.

Lastly, nonrepudiation refers to the inability for a party to dispute or deny having performed an action.

	APNIC	AFRINIC	ARIN	LACNIC	RIPE NCC
Support for delegated RPKI	Yes	No[1]	Yes	Yes[2]	Yes
Publication service for delegated RPKI	Yes	No	Yes	No	Yes
Multi-user support	Yes	Yes	Yes	No	Yes
Two-factor authentication	Yes	Yes	Yes	No	Yes
BGP route collector suggestions	Yes	No	No	Yes	Yes
Auto-renew ROAs	Yes	No	Yes	Yes[3]	Yes
Match "route" objects with ROAs	Yes	No	Yes	No	No
API	No	No	Yes	No	Yes
Nonrepudiation	No	No	Yes	No	No

7.2 Delegated RPKI

Operators who prefer more control and have better integration with their systems can run their own child CA. This model is usually referred to as delegated RPKI.

In this model, the certificate authority that manages object signing is functionally separated from the publication of cryptographic material. This means that an organisation can run a CA and either publish themselves, or delegate this responsibility to a third party, such as a hosting company or cloud provider.

There may be various reasons for organisations to choose this model. For example, this may be useful for organisations that need to be able to delegate RPKI to their customers or different business units, so that that they can run a CA on their systems and manage ROAs themselves.

Alternatively, enterprises who manage large amounts of address space across various RIRs, may not want to manage ROAs in up to five different web interfaces. Instead, they might prefer to be operationally independent from the RIR and manage everything from within one package that is tightly integrated with IP address management and provisioning systems.

Lastly, in the LACNIC and APNIC regions there are several National Internet Registries who provide registration services on a national level to their members and constituents. They also need to be operationally independent and run a certificate authority as a child of their RIR.

[1] Available in the test environment only.
[2] Available upon request.
[3] Explicit opt-in feature.

CHAPTER

EIGHT

USING RPKI DATA

Validation is a key part of any public key infrastructure. The value from signing comes with validation, and should always be done by the party relying on the data. If validation is outsourced to a third party, you can never be certain if the data is complete, or tampered with in any way.

Operators who want to deploy route origin validation in their BGP decision making process have to fetch and validate all of the published RPKI data. As with any PKI, you have to start with one or more entities you are prepared to trust. In the case of RPKI, these are the five Regional Internet Registries.

8.1 Connecting to the Trust Anchor

When you want to retrieve all RPKI data, you connect to the trust anchor that each RIR provides. The root certificate contains pointers to its children, which contain pointers to their children, and so on. These certificates, and other cryptographic material such as ROAs, can be published in the repository that the RIR provides, or a repository operated by an organisation who either runs delegated RPKI themselves, or hosts a repository as a service. As a person who wants to fetch and validate the data, formally known as a relying party, it is not a concern where data is published. By simply connecting to the trust anchor, the chain of trust is followed automatically.

The RIR trust anchor is found through a static trust anchor locator (TAL), which is a very simple file that contains a URL to retrieve the trust anchor and a public key to verify its authenticity. The reason the TAL exists is because it's very likely that the contents of the self signed root certificate change, due to resource transfers between RIRs. By using a TAL, the data in the trust anchor can change, without it needing to be redistributed.

8.2 Fetching and Verifying

Various open source relying party software packages, also known as RPKI validators, are available in order to download, verify and process RPKI data. Please note that most RPKI validators come preinstalled with TALs for all RIRs.

When the validator runs, it will start retrieval at each of the RIR trust anchors and follows the chain of trust to fetch all published certificates and ROAs. Fetching data was originally done via rsync but RIRs and software developers are gradually migrating to the RPKI Repository Delta Protocol (RRDP) for retrieval, standardised in **RFC 8182**. This protocol uses HTTPS, which makes development and implementation easier, and opens up possibilities for Content Delivery Networks to participate in serving RPKI data. Work to deprecate rsync altogether is ongoing in the IETF.

Once the data has been downloaded, the validator will verify the signatures on all objects and output the valid route origins as a list. Each object in this list contains an IP prefix, a maximum length, and an origin AS number. This object is referred to as validated ROA payload (VRP). The collection of VRPs is known as the validated cache.

Note: Objects that do not pass cryptographic verification are discarded. Any statements made about route origins are not considered, as if a ROA was never published. As a result, they will not affect any route announcements.

Please note that objects that do not pass cryptographic verification are sometimes referred to as 'invalid ROAs', but we like to avoid this term because *validity* is used elsewhere in a different context.

Fetching and verification of data should be performed periodically, in order to process updates. Though the standards recommend retrieval at least once every 24 hours, current operational practice recommends that processing updates every 30 to 60 minutes is reasonable.

8.3 Validating Routes

As explained in the *Route Origin Validation* section, when comparing VRPs to the route announcements seen in BGP, it will have an effect on their RPKI validity state. They can be:

Valid
: The route announcement is covered by at least one VRP. The term *covered* means that the prefix in the route announcement is equal, or more specific than the prefix in the VRP.

Invalid
: The prefix is announced from an unauthorised AS, or the announcement is more specific than is allowed by the maxLength set in a VRP that matches the prefix and AS.

NotFound
: The prefix in this announcement is not, or only partially covered by a VRP.

Please carefully note the use of the word *validity*. Because RPKI revolves around signing and verifying cryptographic objects, it's easy to confuse this term with the validity state of a BGP announcement. As mentioned, it can occur that a ROA doesn't pass cryptographic verification, for example because it expired. As a result, it is discarded and will not affect any BGP announcement. In turn, only a validated ROA payload—sometimes referred to as 'valid ROA'—can make a BGP announcement Valid or Invalid.

A route announcement may be covered by several VRPs. For example, there may be a VRP for the aggregate announcement, which overlaps with a customer announcement of a more specific prefix from a different AS. A route announcement will be Valid as long as there is one covering VRP that authorises it.

Based on the three validity outcomes, operators can make an informed decision what to do with the BGP route announcements they see. As a general guideline, announcements with Valid origins should be preferred over those with NotFound or Invalid origins. Announcements with NotFound origins should be preferred over those with Invalid origins.

As origin validation is deployed incrementally, the amount of IP address space that is covered by a ROA will gradually increase over time. Therefore, accepting the NotFound validity should be done for the foreseeable future.

Important: For route origin validation to succeed in its objective, operators should ultimately drop all BGP announcements that are marked as Invalid. Before taking this step, organisations should first analyse the effects of doing this, to avoid unintended results. Initially accepting Invalid announcements and giving them a lower preference, as well as tagging them with a BGP community is a good first step to measure this.

8.4 Local Overrides

Sometimes there is an operational need to accept Invalid announcements temporarily. Local overrides allow you to manage your own exceptions to the validated cache. This ensures that you remain in full control of the VRPs used by your routers. For example, if an Invalid origin is the result of a misconfigured ROA, you may accept it until the operator in question has resolved the issue. A format named SLURM is available for this, which is standardised in **RFC 8416**.

SLURM provides several ways to achieve exceptions. First, you can add a VRP specifically for the affected route by specifying the correct ASN, prefix and maximum length. Secondly, you can filter out an existing VRP, thereby moving the route back to NotFound state. In general, the former is the safer way, as it deals better with changing ROAs. Lastly, it is possible to allow all routes from a certain ASN or prefix. It is advised to use overrides with care, as liberal usage may have unintended consequences.

8.5 Feeding Routers

The validated cache can be fed directly into RPKI-capable routers via the RPKI to Router Protocol (RPKI-RTR), described in **RFC 8210**. Many routers, including Cisco, Juniper, Nokia, as well as BIRD and OpenBGPD support processing the validated cache. Alternatively, most validators can export the cache in various useful formats for processing outside of the router, in order to set up filters.

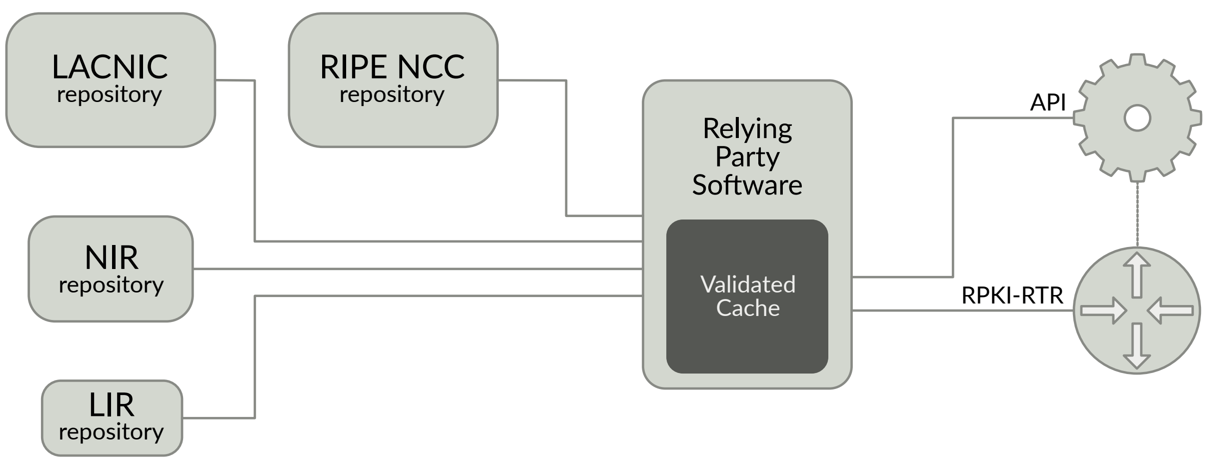

Fig. 8.1: RPKI publication, data retrieval, validation and processing

Note that your router does not perform any of the cryptographic validation, this is all handled by the relying party software. In addition, using RPKI causes minimal overhead for routers and has a negligible influence on convergence speed. Validation happens in parallel with route learning for new prefixes which are not yet in the cache. Those prefixes will be marked as Valid, Invalid, or NotFound as the information becomes available, after which the correct policy is applied.

Please keep in mind that the RPKI validator software you run in your network fetches cryptographic material from the outside world. To do this, it needs at least ports 873 and 443 open for rsync and HTTPS, respectively. In most cases, the processed data is fed to a router via RPKI-RTR over a clear channel, as it's running in your local network. Currently, only Cisco IOS-XR provides a practical means to secure transports for RPKI-RTR, using SSH.

It is recommended to run multiple validator instances as a failover measure. The router will use the union of RPKI data from all validators to which they are connected. This means that (temporary) differences in the validated cache produced by the validators, for example due to differing fetching intervals, does not pose a problem.

In the *Router Support* section we will look at which routers support route origin validation, and how to get started with each.

CHAPTER

NINE

SOFTWARE PROJECTS

This section provides an overview of all well known open source projects that support RPKI. It includes Relying Party software for validating RPKI data, Certificate Authority software to run RPKI on your own infrastructure and supporting tools that help deployment and integration.

9.1 Relying Party Software

Name	Maintainer	Language	Last Commit
FORT Validator	NIC.mx	C	
OctoRPKI	Cloudflare	Go	
rcynic	Dragon Research Labs	Python 2	
Routinator	NLnet Labs	Rust	
rpki-client	OpenBSD	C	
rpki-prover	Misha Puzanov	Haskell	
RPSTIR2	ZDNS	Go	

9.2 RTR Server Software

Name	Maintainer	Language	Last Commit
GoRTR[1]	Cloudflare	Go	
StayRTR[2]	bgp	Go	
RTRTR	NLnet Labs	Rust	
rpkirtr	Darren O'Connor	Go	

[1] Unmaintained since the developer got a new job. [Source]

[2] A fork of GoRTR

9.3 Certificate Authority Software

Name	Maintainer	Language	Last Commit
Krill	NLnet Labs	Rust	
rpkid	Dragon Research Labs	Python 2	

9.4 Supporting Tools

BGPalerter
A self-configuring BGP monitoring tool, which allows you to monitor in real-time if any of your prefixes loses visibility or is hijacked, your AS is announcing RPKI invalid prefixes or is announcing prefixes not covered by ROAs, ROAs covering your prefixes are no longer reachable, and much more.

BGP-SRx
SRx is an open source reference implementation and research platform by the National Institute for Standards and Technology (NIST). It is intended for investigating emerging BGP security extensions and supporting protocols such as RPKI Origin Validation and BGPSec Path Validation.

krill-sync
This tool uses the RRDP data from a (single) "hidden" backend RPKI Publication Server to make a consistent local copy of that data. This is intended to facilitate a redundant set up where one or more public https and rsync servers are used to make the RPKI repository content available.

pmacct
pmacct is a small set of multi-purpose passive network monitoring tools. It can account, classify, aggregate, replicate and export forwarding-plane data, i.e. IPv4 and IPv6 traffic; collect and correlate control-plane data via BGP and BMP; collect and correlate RPKI data; collect infrastructure data via Streaming Telemetry.

The pmacct toolset can perform RPKI Origin Validation and present the outcome as a property in the flow aggregation process. Because it separates out the various types kinds of (invalid) BGP announcements, operators can a good grasp on how their connectivity to the rest of the Internet would look like after deploying a *"invalid == reject"* policy.

rpki-ov-checker
rpki-ov-checker is an open source utility to quickly analyse BGP RIB dumps and the potential impact of deploying "invalid is reject" routing policies.

RTRLib
The RTRlib implements the client-side of the RPKI-RTR protocol (**RFC 6810**, **RFC 8210**) and BGP Prefix Origin Validation (**RFC 6811**). This also enables the maintenance of router keys, which are required to deploy BGPSec.

CHAPTER

TEN

ROUTER SUPPORT

Several router vendors participated in the development of the RPKI standards in the IETF, ensuring the technology offered an end-to-end solution for route origin validation. The RPKI to Router protocol (RPKI-RTR) is standardised in **RFC 6810** (v0) and **RFC 8210** (v1). Is it specifically designed to deliver validated prefix origin data to routers. This, as well as origin validation functionality, is currently available in on various hardware platforms and software solutions.

10.1 Hardware Solutions

Important: The versions listed here are the earliest ones where RPKI support became available. However, a newer version may be required to get recommended improvements and bug fixes. Please check your vendor documentation and knowledge base.

Juniper — Documentation
Junos version 12.2 and newer. Please read PR1461602 and PR1309944 before deploying.

Cisco — Documentation
IOS release 15.2 and newer, as well as Cisco IOS/XR since release 4.3.2.

Nokia — Documentation
SR OS 12.0.R4 and newer, running on the 7210 SAS, 7250 IXR, 7750 SR, 7950 XRS and the VSR.

Arista — Blog post
EOS 4.24.0F and newer

MikroTik — Documentation
7.1 and newer

Huawei - Documentation
VRP 8.150 and newer.

10.2 Software Solutions

Various software solutions have support for origin validation:

- BIRD
- OpenBGPD
- FRRouting
- GoBGP

- VyOS

In some solutions, such as OpenBGPD, RPKI-RTR is not available but the same result can be achieved through a static configuration. The router will periodically fetch the validated cache and allow operators to set up route maps based on the result. *Relying party software* such as Routinator and rpki-client can export validated data in a format that OpenBGPD can parse.

RTRLib is a C library that implements the client side of the RPKI-RTR protocol, as well as route origin validation. RTRlib powers RPKI in BGP software routers such as FRR. In a nutshell, it maintains data from RPKI relying party software and allows to verify whether an autonomous system (AS) is the legitimate origin AS, based on the fetched valid ROA data. BGP-SRx by NIST is a prototype that can perform similar functions.

CHAPTER

ELEVEN

RESOURCES

This page provides an overview of projects that support RPKI. It includes, statistics, measurements projects and presentations about operational experiences. Finally, there is an overview of all work in the Internet Engineering Task Force relevant to RPKI.

The *Software Projects* page an overview of all available tools for using RPKI.

11.1 Books

`BGP RPKI: Instructions for use`, by Flavio Luciani & Tiziano Tofoni (PDF)

`Juniper Day One: Deploying BGP Routing Security`, by Melchior Aelmans & Niels Raijer (PDF)

11.2 Insights and Statistics

There are several initiatives that measure the adoption and data quality of RPKI:

- Cirrus Certificate Transparency Log, by Cloudflare
- Global certificate and ROA statistics, by RIPE NCC
- RPKI Deployment Monitor, by NIST
- The RPKI Observatory, by nusenu
- RPKI connection test, by RIPE Labs
- The rpki-client console, by the OpenBSD project
- JDR: explore, inspect and troubleshoot anything RPKI, by NLnet Labs
- RPKIviews: download historic raw RPKI data, by Job Snijders
- ROA Use World Map and Country Data, by APNIC Labs
- RPKI Portal, by Cloudflare
- LACNIC Member Tools, by LACNIC

11.3 Operational Experiences

RPKI Deployment Considerations for ISPs
Document by Rich Compton - Charter Communications, with contributions from the operator community

Using RPKI with IXP Manager
Documentation to set up Routinator, OctoRPKI and the RIPE NCC Validator with BIRD 2.x

Use Routinator with Cisco IOS-XR
Blog post by Fabien Vincent

Wikimedia RPKI Validation Implementation
Documentation by Arzhel Younsi describing RPKI validator and router configuration

Dropping RPKI invalid routes in a service provider network
Lightning talk by Nimrod Levy - AT&T, NANOG 75, February 2019

RPKI and BGP: our path to securing Internet Routing
Blog post by Jérôme Fleury & Louis Poinsignon - Cloudflare, September 2018

RPKI For Managers
Presentation by Niels Raijer - Fusix Networks, NLNOG Day 2018, September 2018

RPKI at IXP Route Servers
Presentation by Nick Hilliard - INEX, RIPE 78, May 2019

Lessons learned: NTT's RPKI deployment
Presentation by Job Snijders - NANOG 79, June 2020

11.4 Examples of BGP Hijacks

How Verizon and a BGP Optimizer Knocked Large Parts of the Internet Offline Today
Cloudflare Blog, 24 June 2019

BGP / DNS Hijacks Target Payment Systems
Oracle Internet Intelligence, 3 August 2018

Shutting down the BGP Hijack Factory
Oracle Dyn, 10 July 2018

Suspicious event hijacks Amazon traffic for 2 hours, steals cryptocurrency
Ars Technica, 24 April 2018

Popular Destinations rerouted to Russia
BGPmon, 12 December 2017

Insecure routing redirects YouTube to Pakistan
Ars Technica, 25 February 2008

11.5 IETF Documents

Most of the original work on RPKI standardisation for both origin and path validation was done in the Secure Inter-Domain Routing (sidr) working group. After the work was completed, the working group was concluded.

Since then, the SIDR Operations (sidrops) working group was formed. This working group develops guidelines for the operation of SIDR-aware networks, and provides operational guidance on how to deploy and operate SIDR technologies in existing and new networks.

All relevant drafts and standards can be found in the archives of these two working groups as well as the RPKI RFCs Graph.

CHAPTER

TWELVE

INDEX

- genindex

INDEX