RIPE NCC
RIPE NETWORK COORDINATION CENTRE

BGP Routing Security Workshop

RIPE NCC Learning and Development

Jad El Cham | May 2024 | RIPE 88 - Krakow

Overview

- RPKI Refresher
  - Vulnerabilities of the BGP Protocol
  - Introducing RPKI
- RPKI Validation
  - Running a Validator
  - Filtering with BGP OV
- Next steps for BGP Security

RPKI Refresher

Vulnerabilities of the BGP Protocol

BGP Has Some Challenges

- It is only based on **trust**, no built-in security
- **No verification** of how correct prefixes or AS paths are

RFC 4272 - “BGP Security Vulnerabilities Analysis”

BGP Has Three Main Vulnerabilities

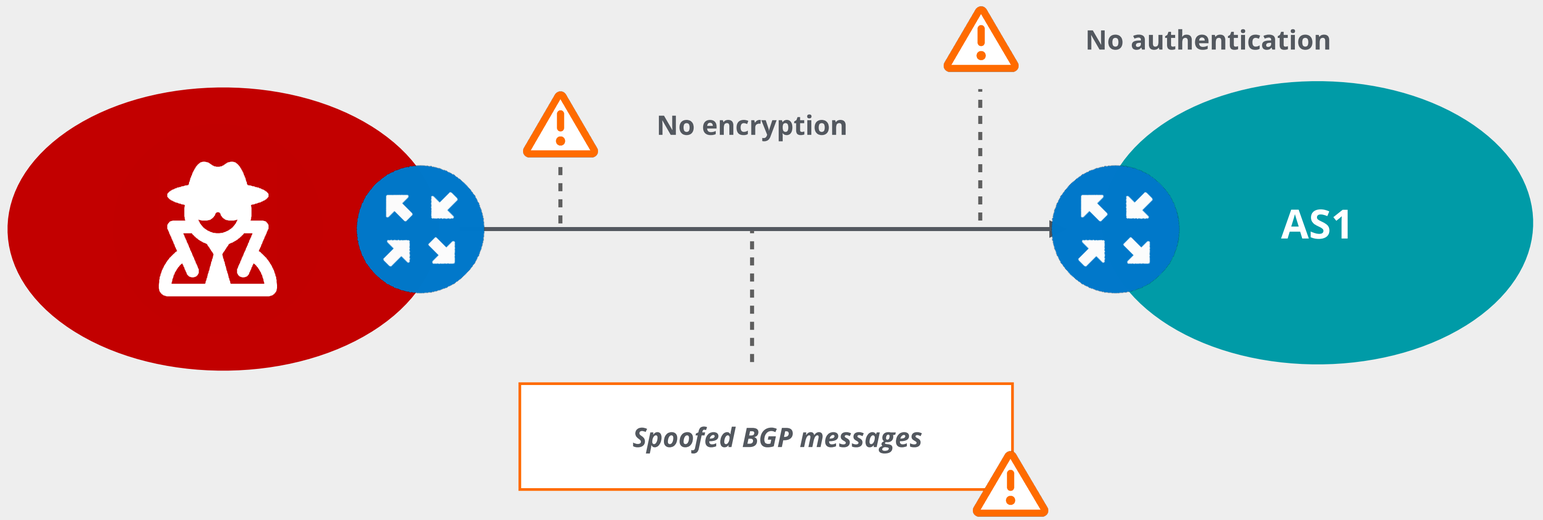
1. No internal mechanism to protect the integrity and source authenticity of BGP messages, and no confidentiality

2. No mechanism specified to validate the authority of an AS to announce a prefix

3. No mechanism to verify the authenticity of the attributes in a BGP update message

No Encryption or Authentication

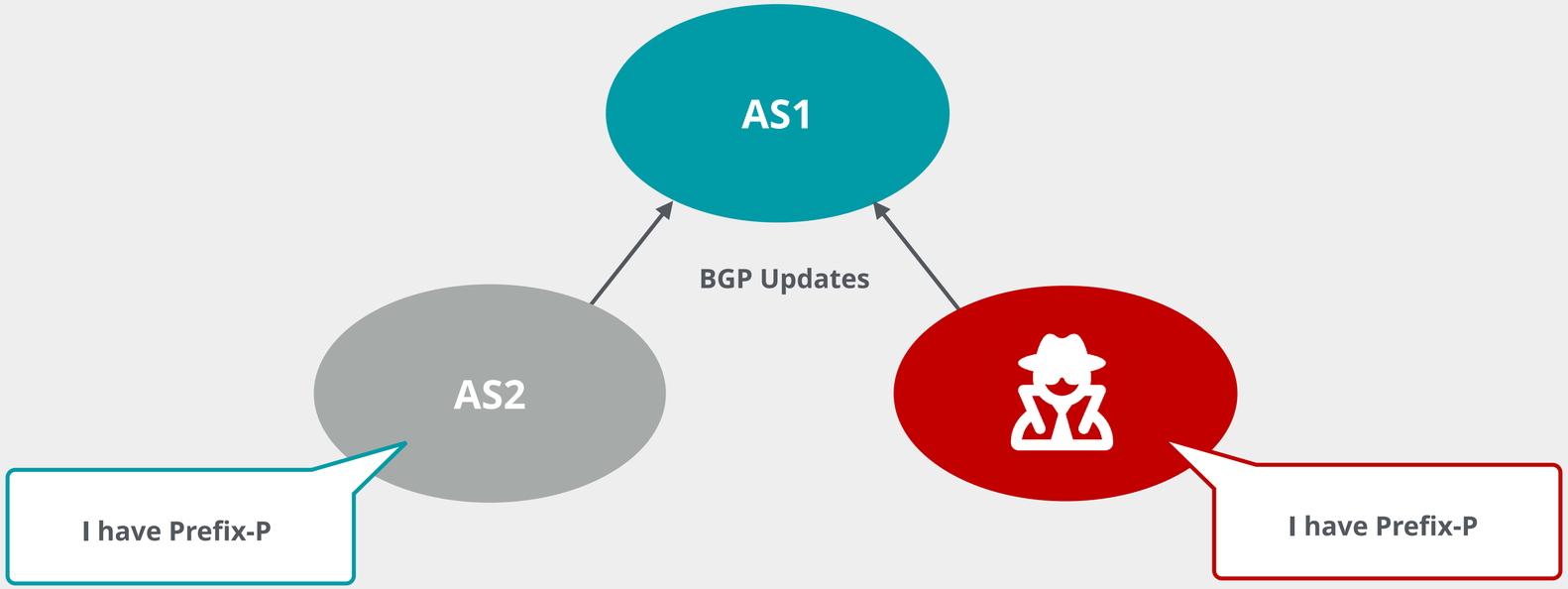
- BGP **does not** have a built-in authentication mechanism
- BGP provides **no integrity** or **confidentiality**
- BGP messages do not use freshness service and can be replayed



No Origin Validation

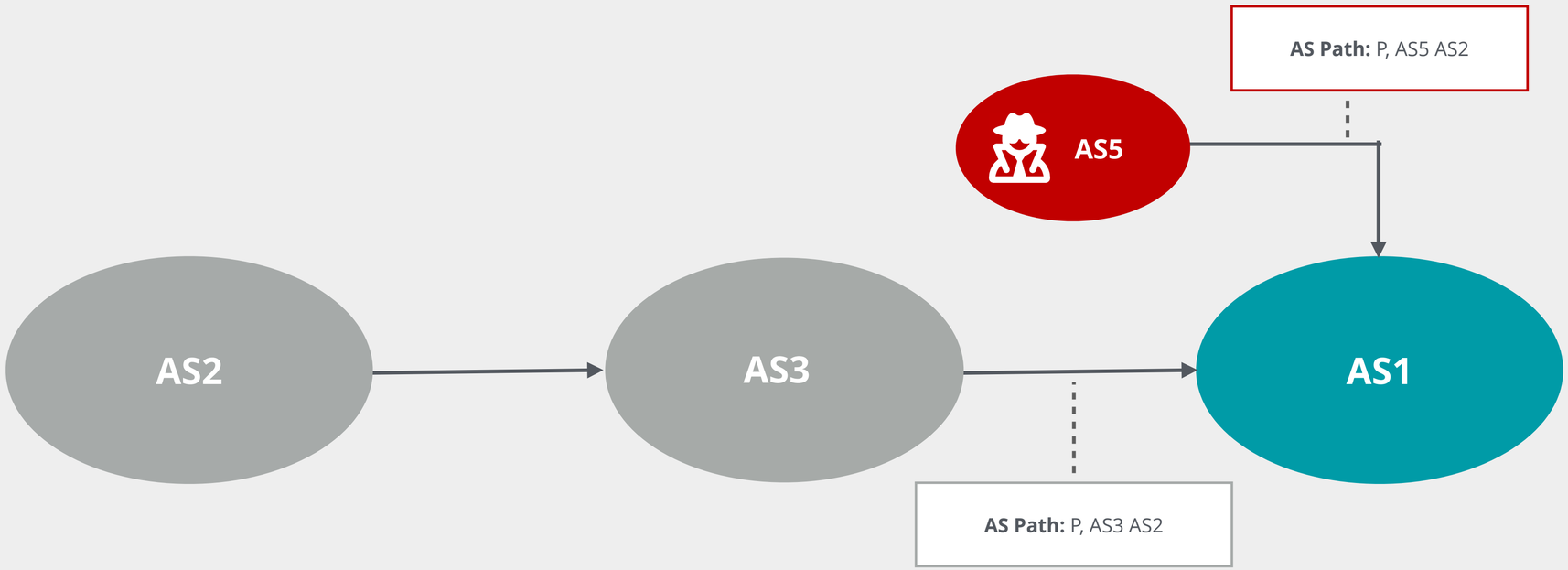
- BGP does not have a validity check for propagated routes

  - Any AS can announce any prefix!



No Authentication of AS Path

- AS path attribute received in BGP update cannot be validated
- Anyone can alter the path and prepend any ASN to the AS path

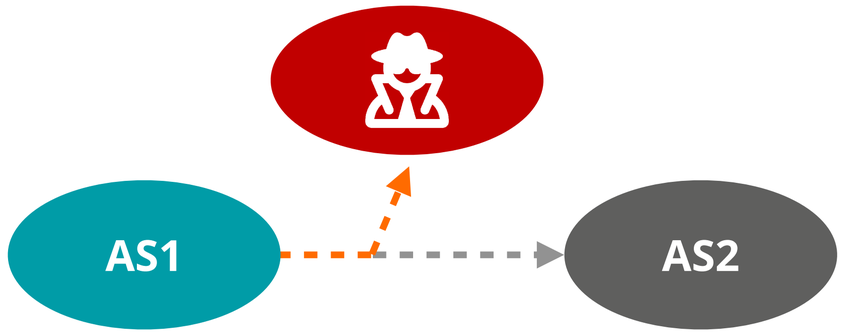


Due To These Vulnerabilities

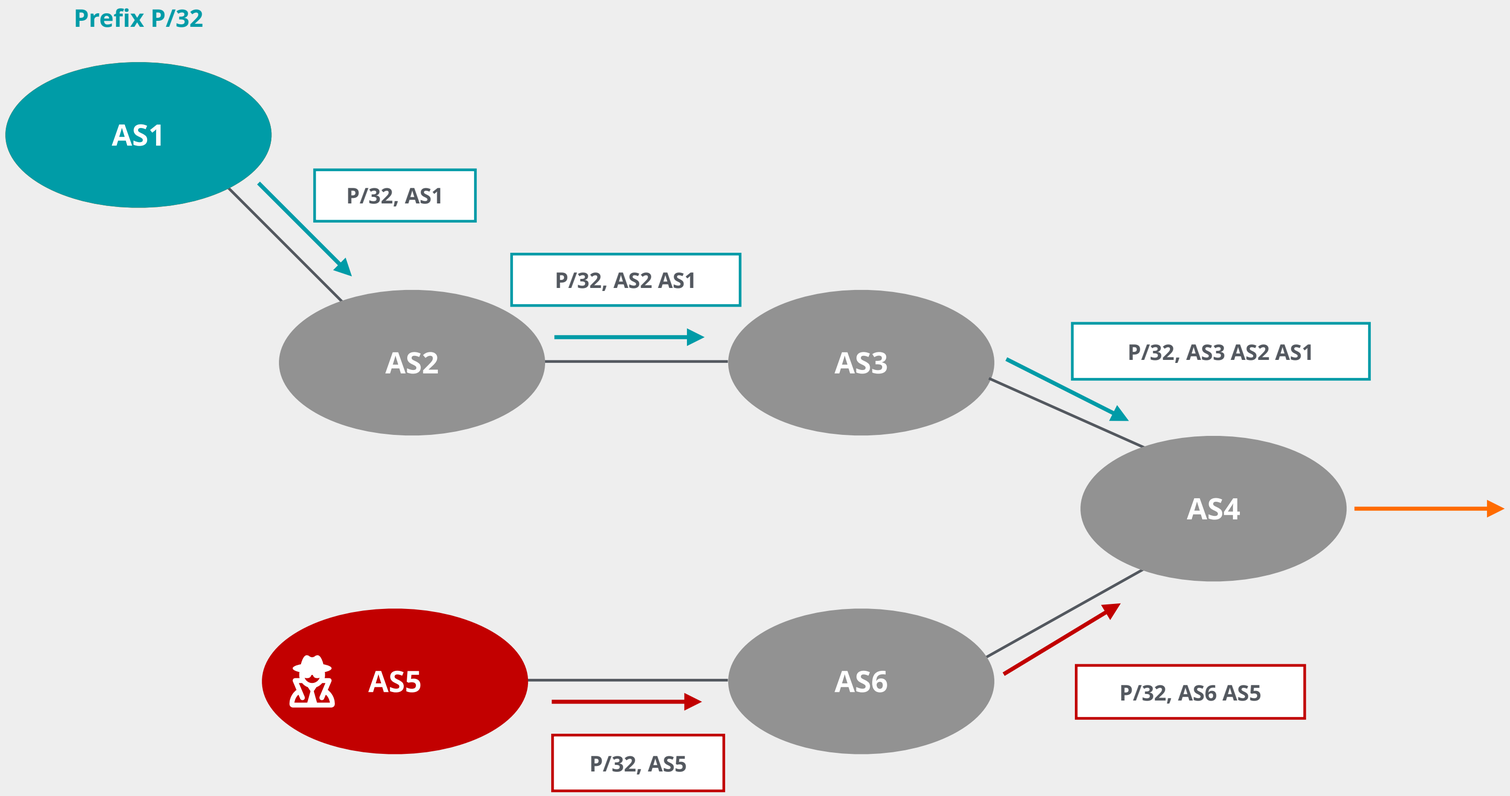
- Attacks can be conducted by exploiting TCP or BGP messages
- Any AS can announce any prefix
- Any AS can prepend any ASN to the AS path
- Fake routing information may disrupt Internet routing

BGP Origin Hijacks

- An AS originates a prefix that is not authorised to originate
- Hijacker impersonates the legitimate holder
  - May hijack an **allocated** or **unallocated** address space
- It may announce the exact same prefix or more specifics
  - **Prefix Hijack**
  - **Sub-prefix Hijack** (De-aggregation hijack or subnet attack)



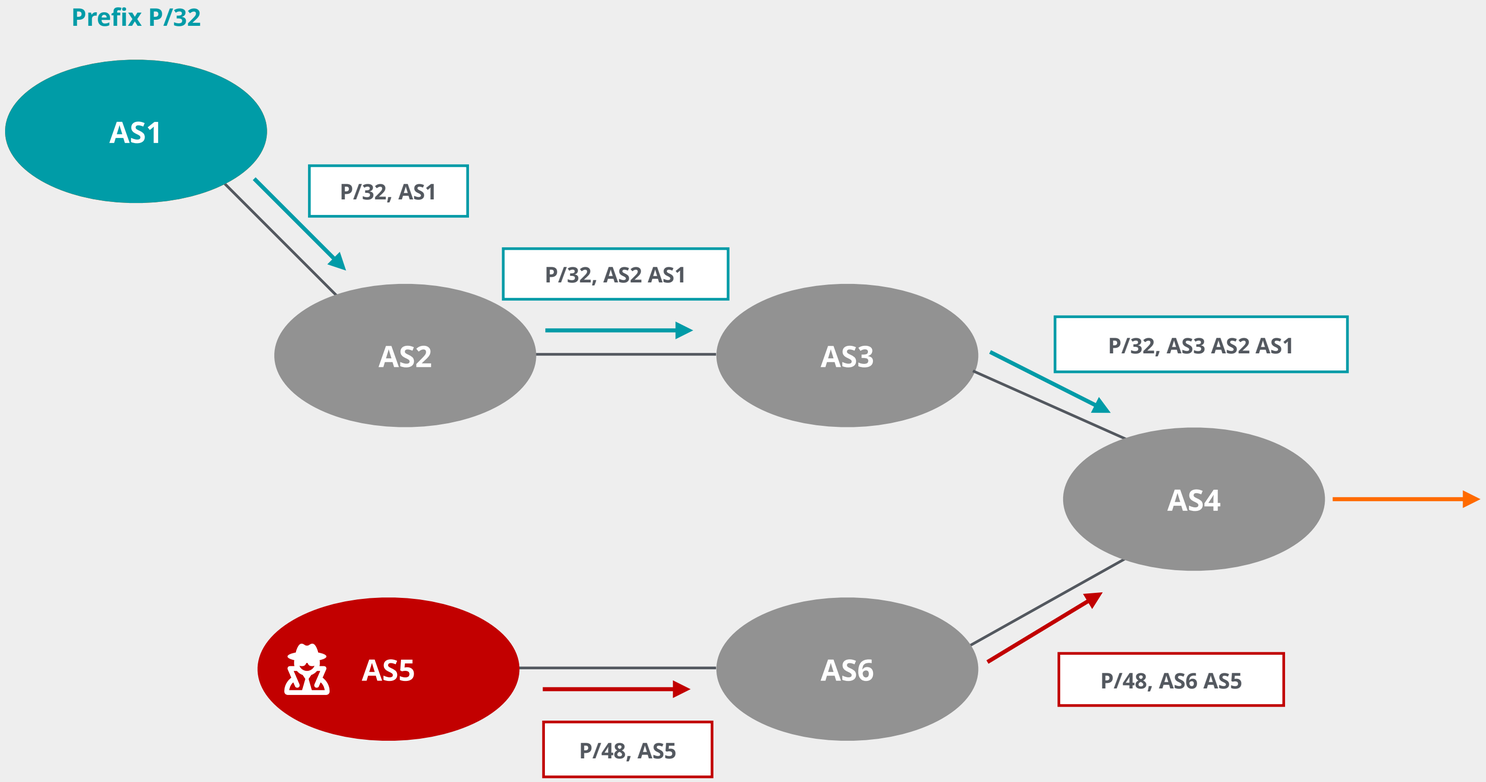
Prefix Hijack



This is a **local hijack!**

Only some networks are affected based on BGP path selection process

Sub-prefix Hijack (Subnet Attack)



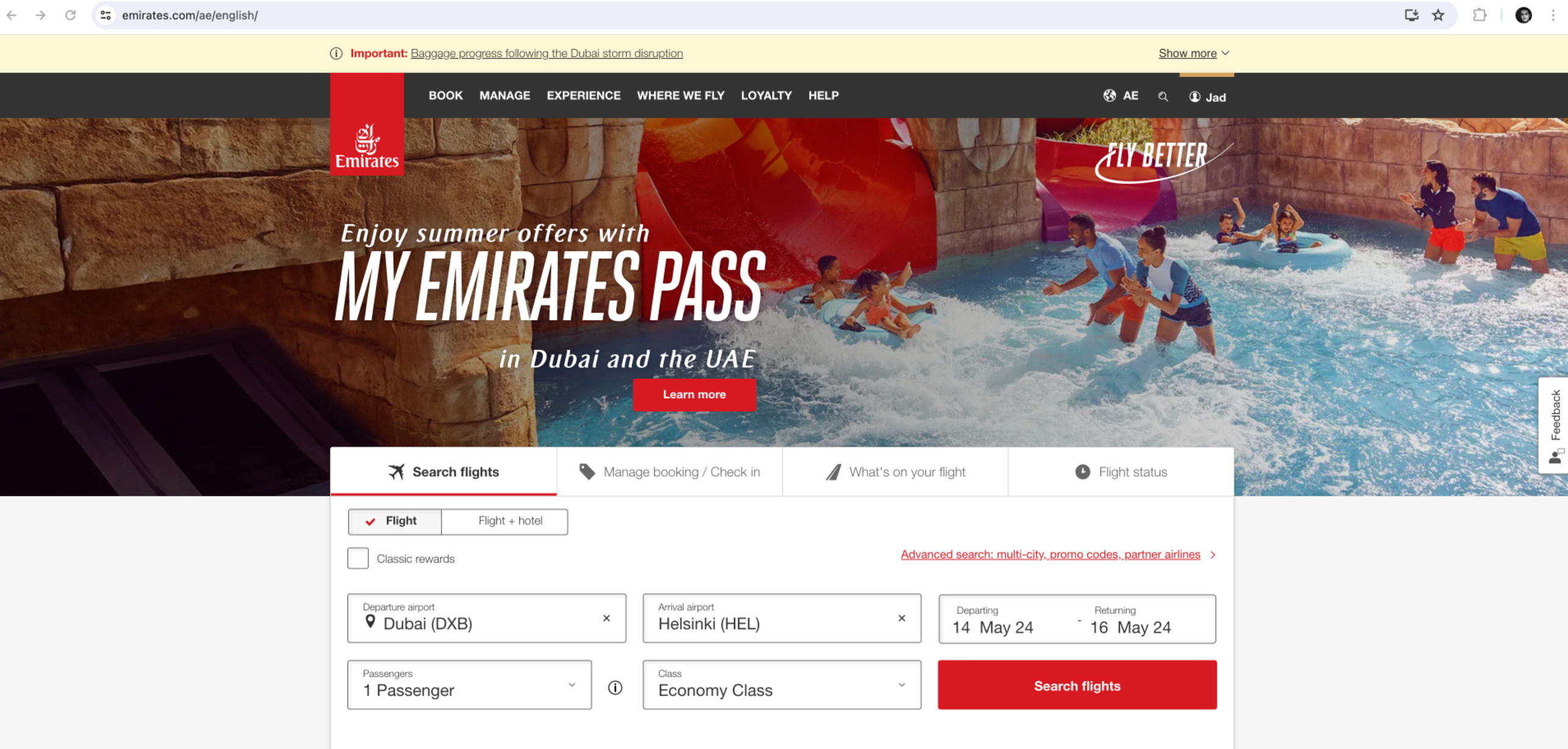
This is a **global hijack!**

All traffic for more specific prefix will be forwarded to the hijacker's network

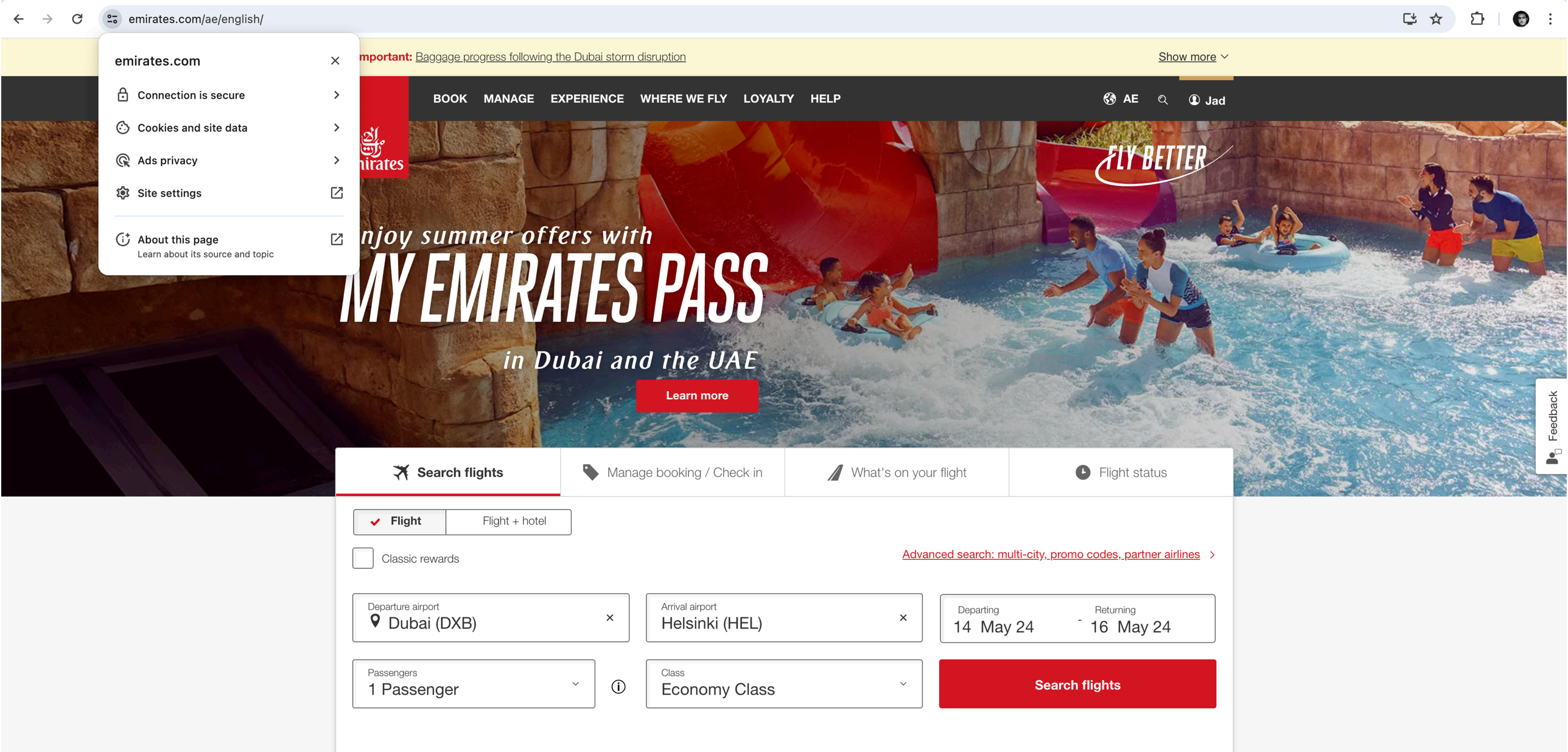
Introducing RPKI

But wait… Let’s book some flight tickets!

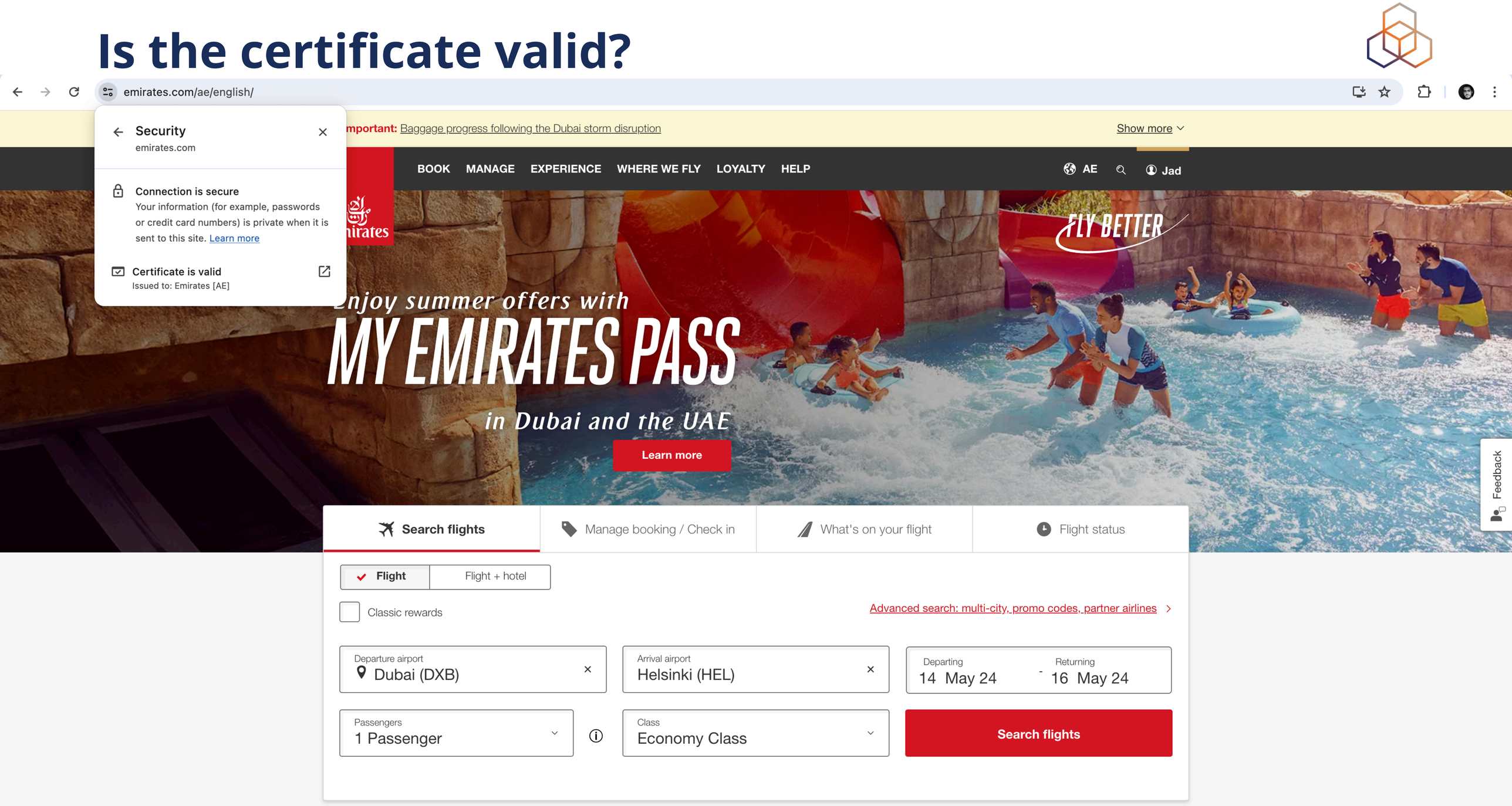
Can you trust this website?



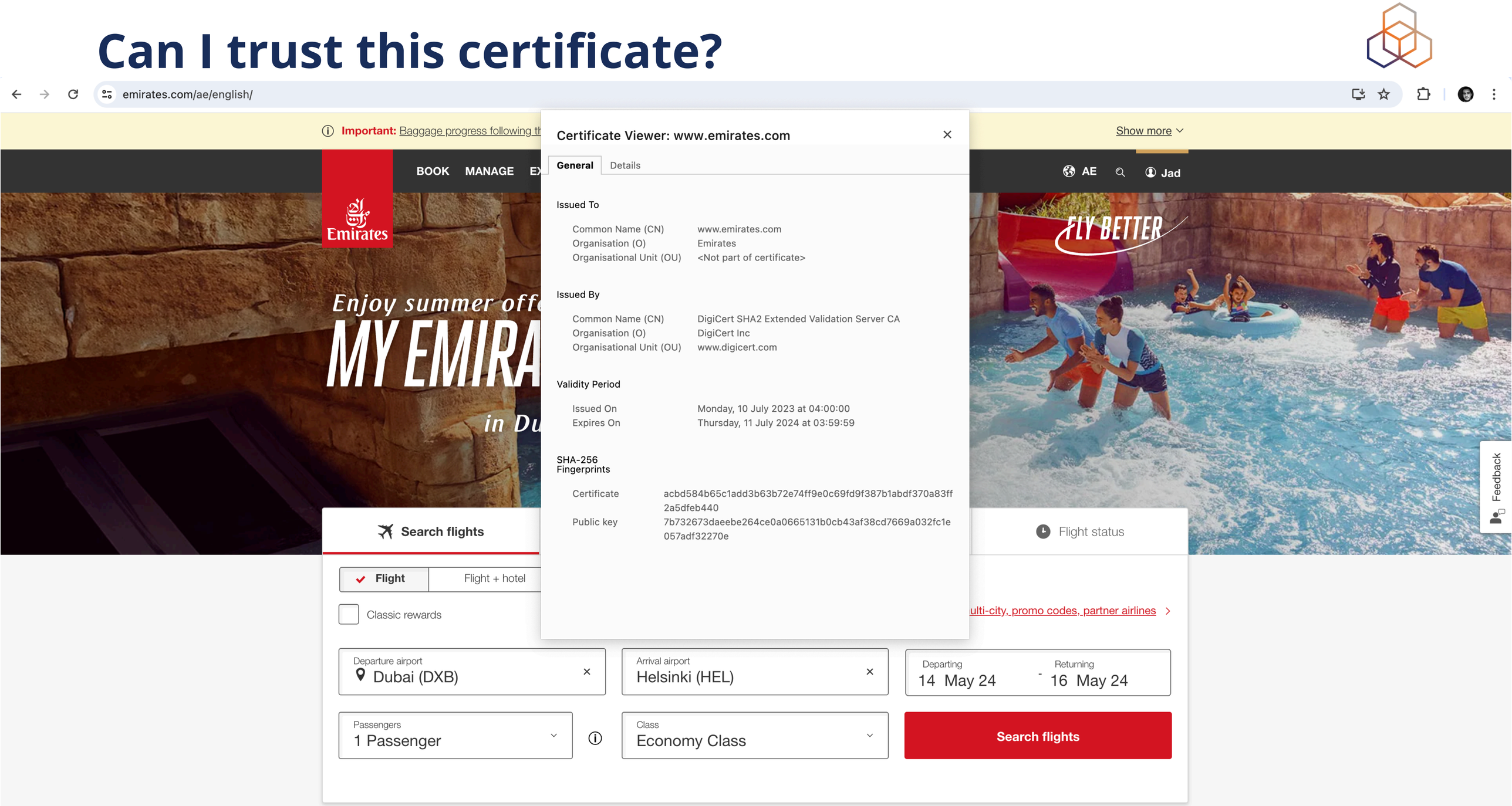
The connection seems secure



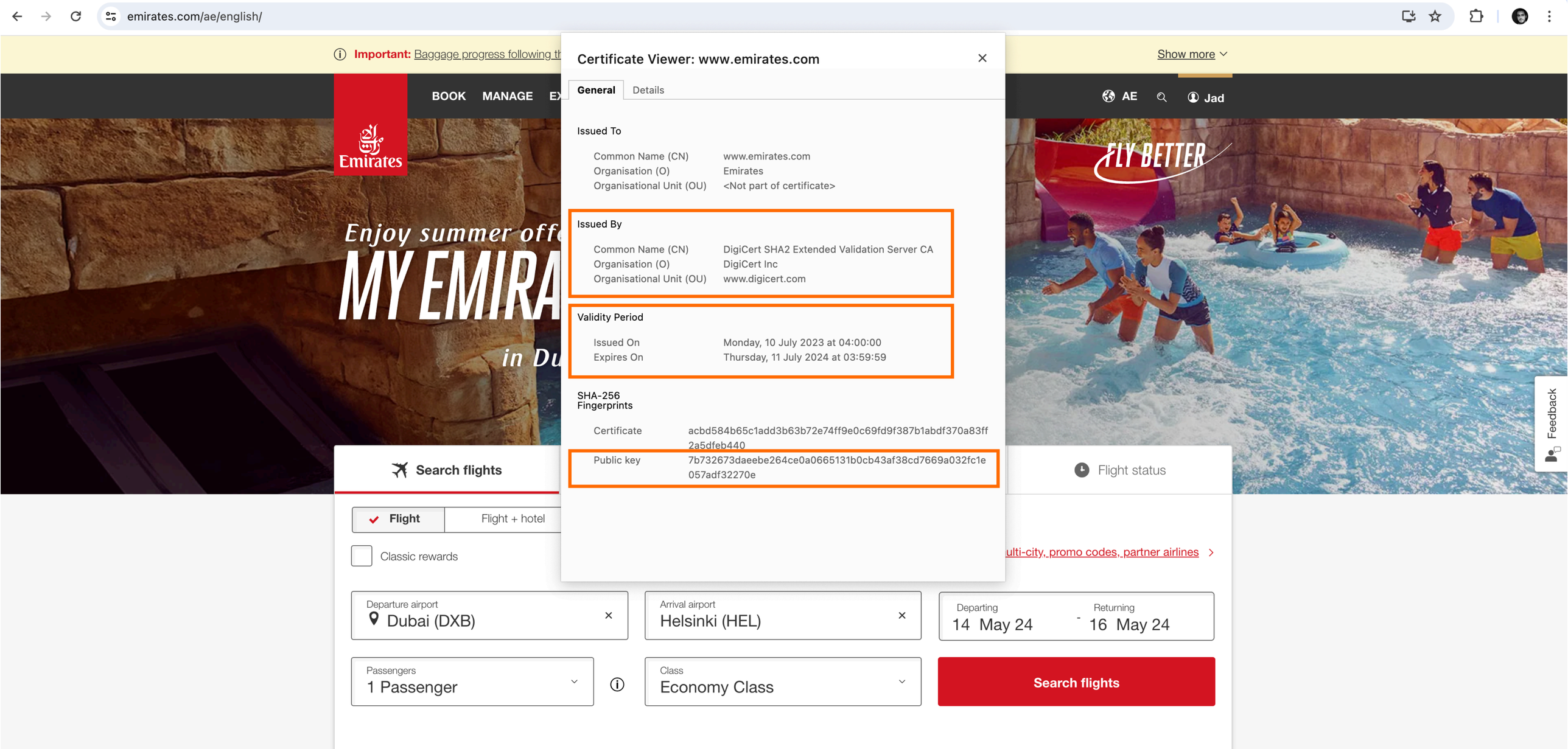
Is the certificate valid?



Can I trust this certificate?



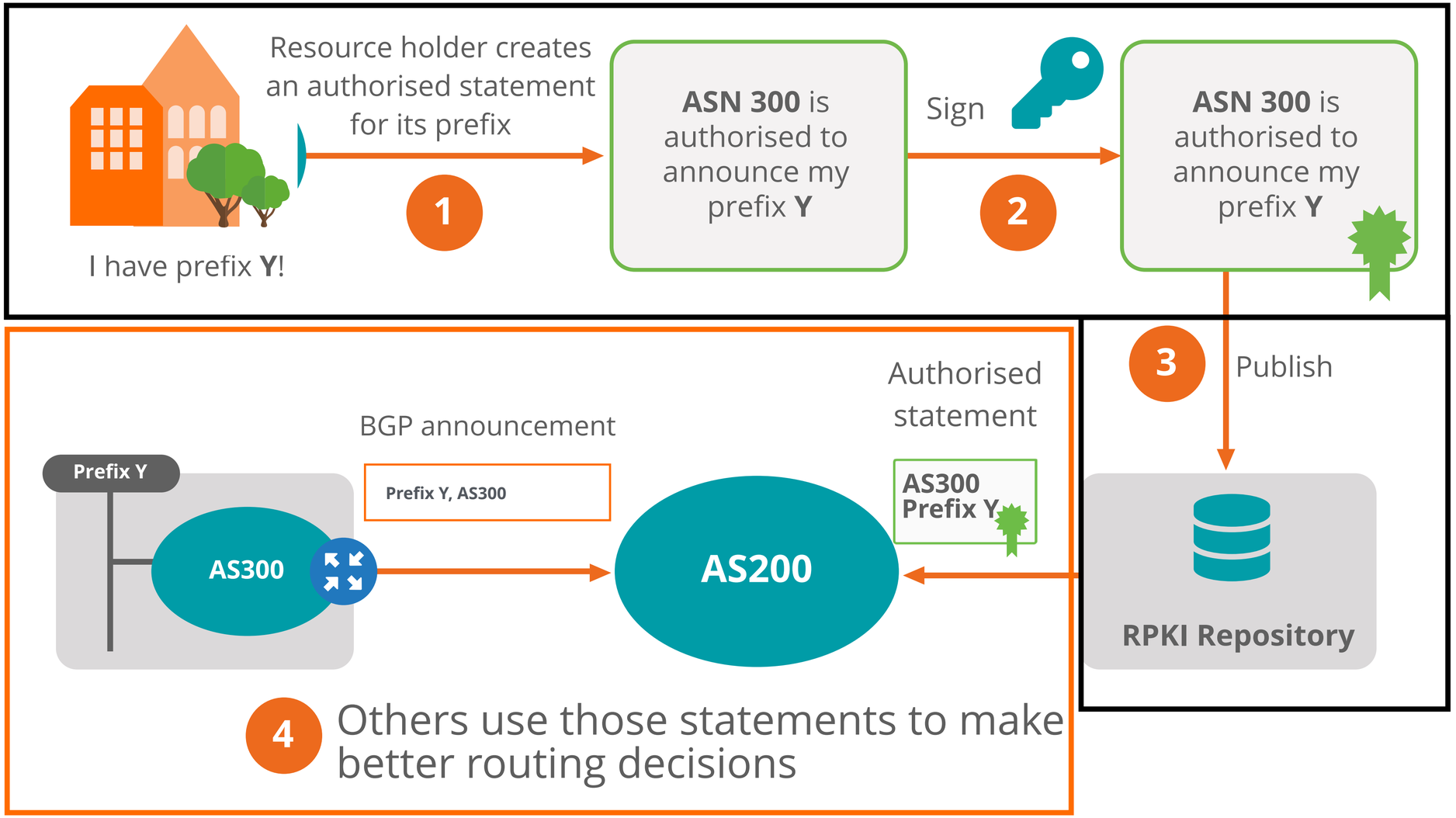
Oh yes, I trust the issuer!



What is RPKI?

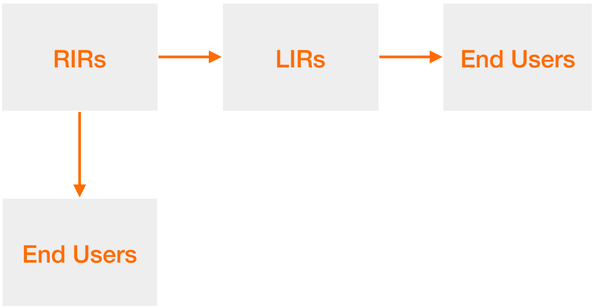
- A security framework for the Internet
- **Verifies the association between resource holders and their resources**
  - Attaches digital certificate to IP addresses and AS numbers
- Used to **validate the origin** of BGP announcements (BGP OV)
  - Is the originating ASN authorised to originate a particular prefix?
  - Helps to mitigate BGP Origin Hijacks and Route leaks

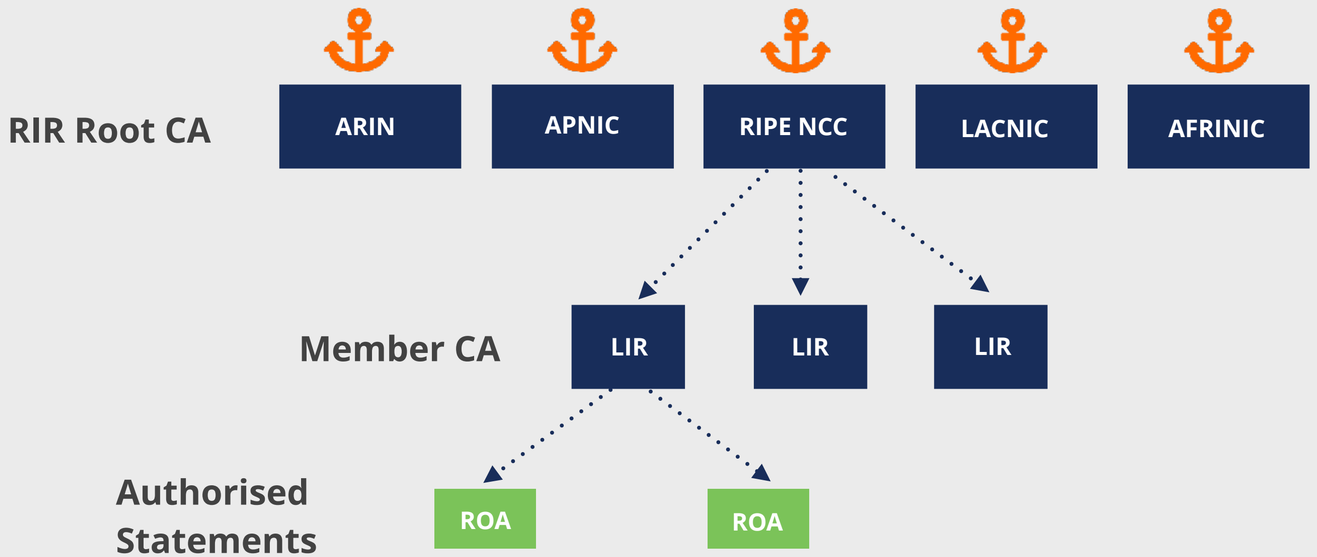
How Does RPKI Work?



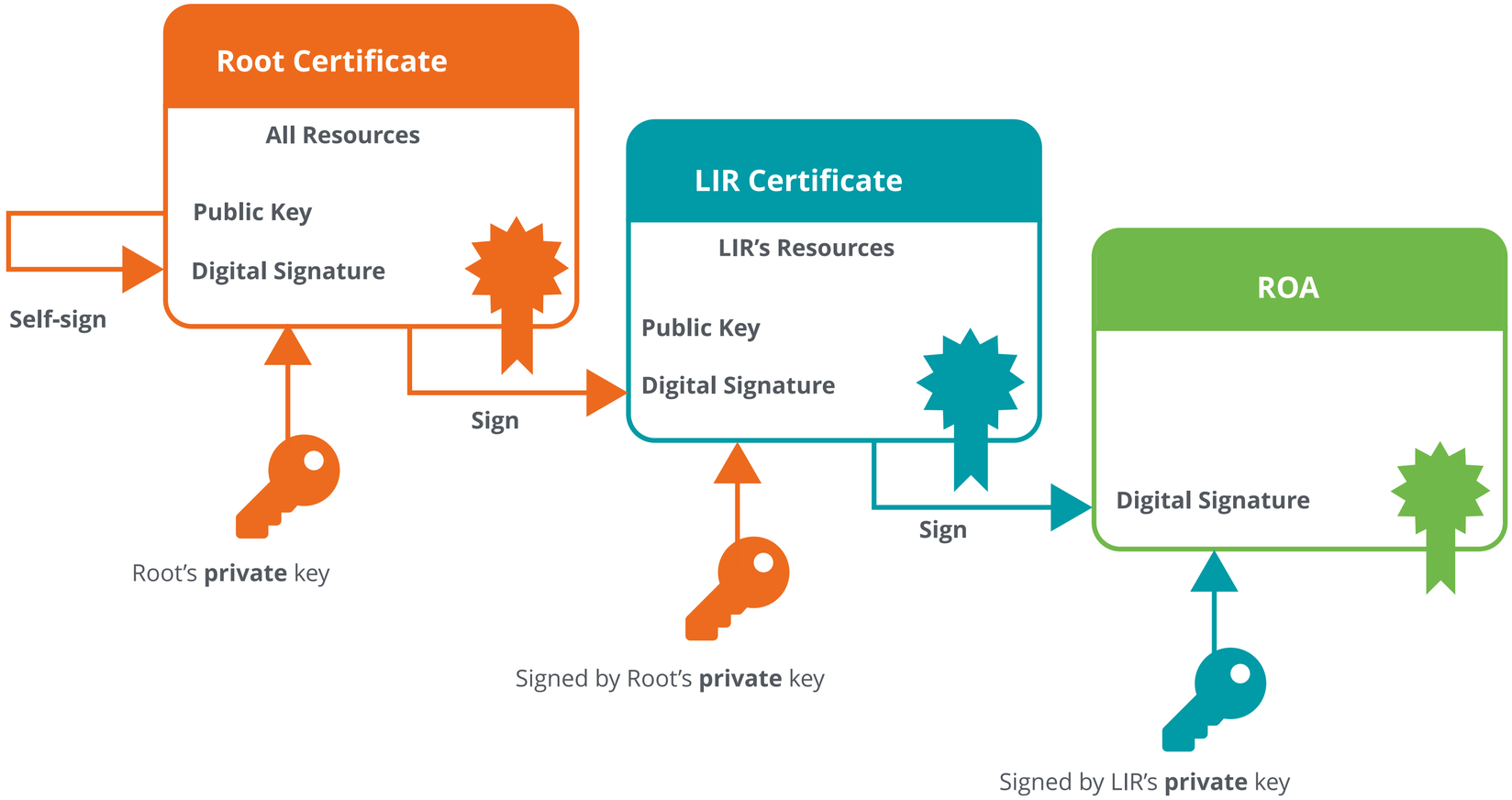
Trust in RPKI

- RPKI relies on five RIRs as Trust Anchors
- Certificate structure follows the RIR hierarchy
- RIRs issue certificates to resource holders





RPKI Chain of Trust



Elements of RPKI

- The RPKI system consists of two parts:

SIGNING

Create ROAs for your prefixes in the RPKI system

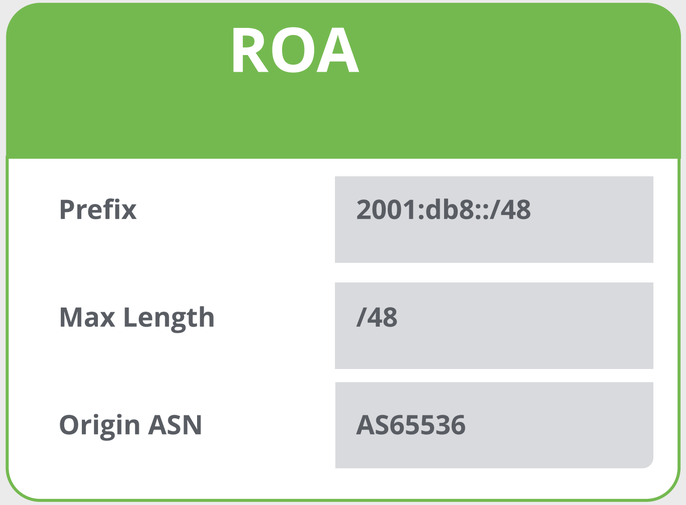
+

VALIDATION

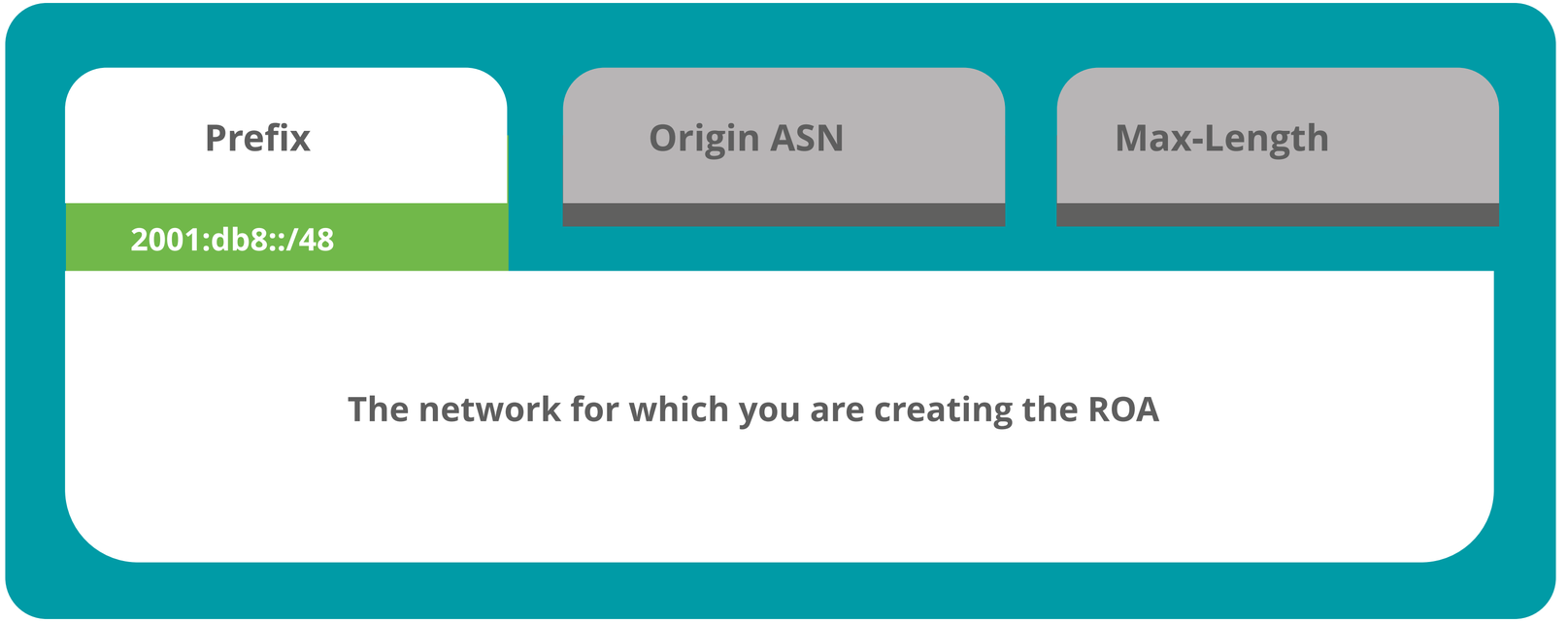
Verify the information provided by others

What is a ROA?

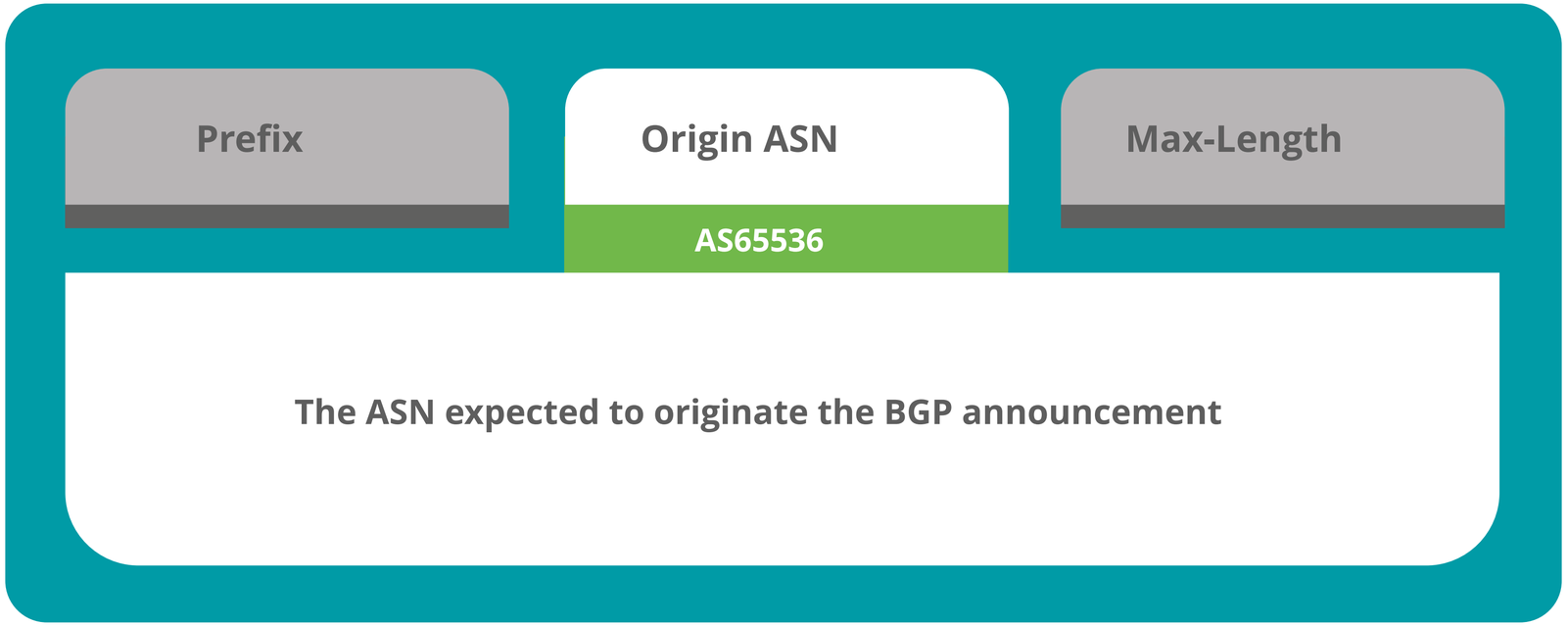
- An **authorised statement** from a resource holder
  - States that a certain prefix can be originated by a certain AS
- Contains a list of IP address prefixes and an AS number
- Multiple ROAs can exist for the same prefix
- ROAs can overlap



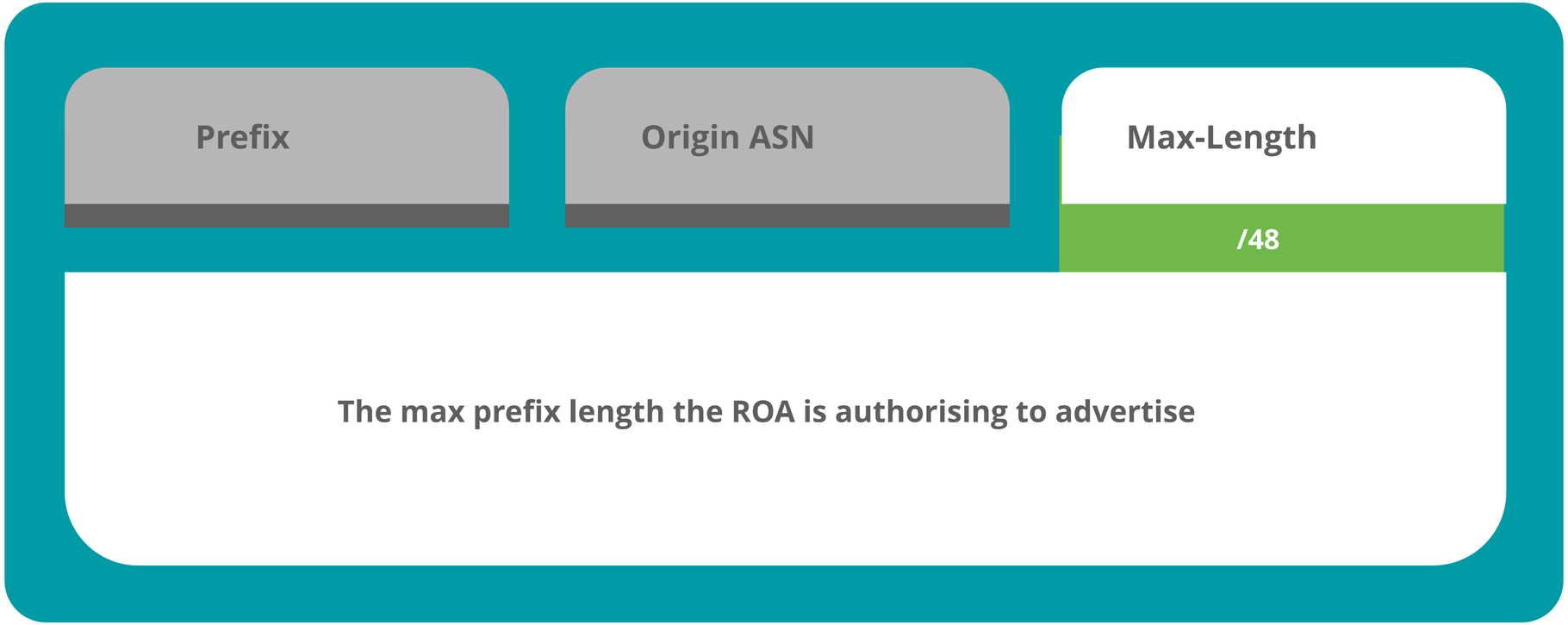
What is in a ROA?



What is in a ROA?

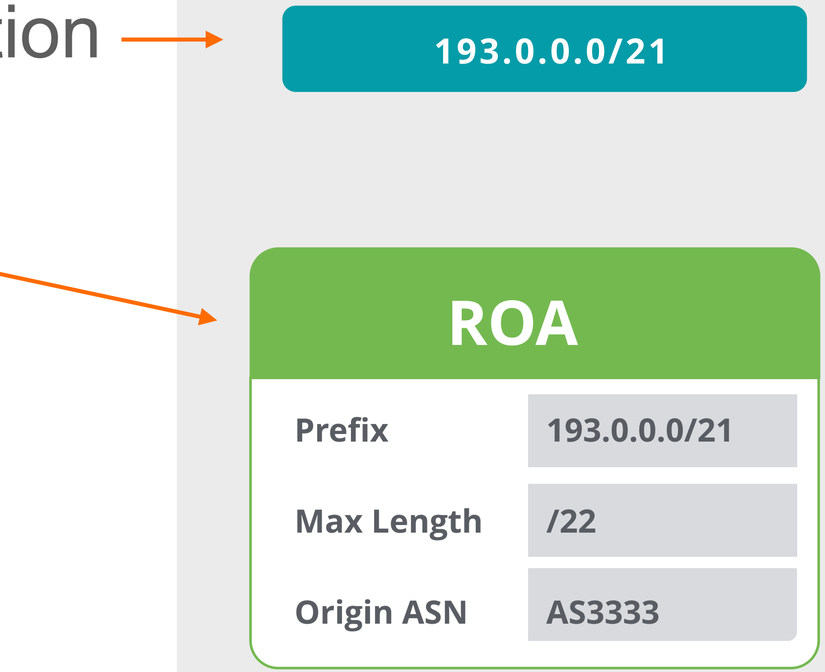


What is in a ROA?



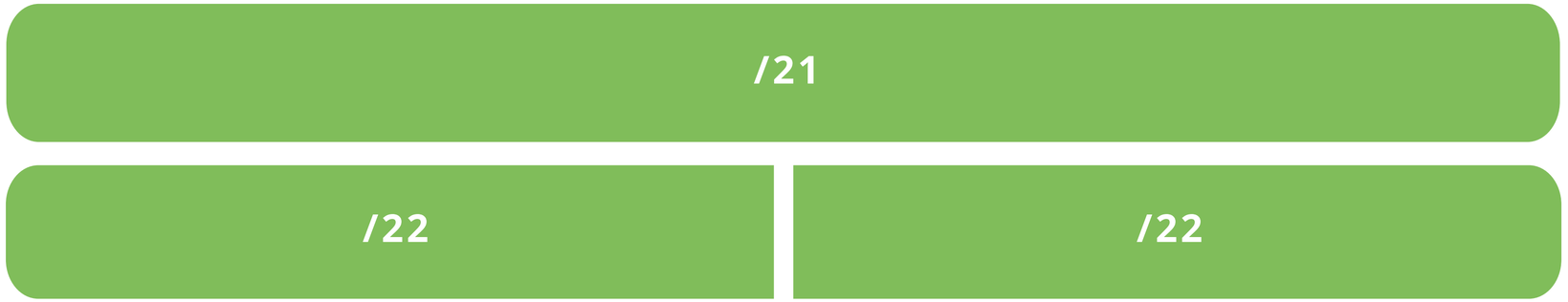
Max-Length

- RIPE NCC (AS3333) has an IP address allocation
- RIPE NCC creates this ROA



Max-Length

- RIPE NCC (AS3333) has an IP address allocation
- RIPE NCC creates this ROA
- According to the ROA:



193.0.0.0/21

ROA

Prefix	193.0.0.0/21
Max Length	/22
Origin ASN	AS3333

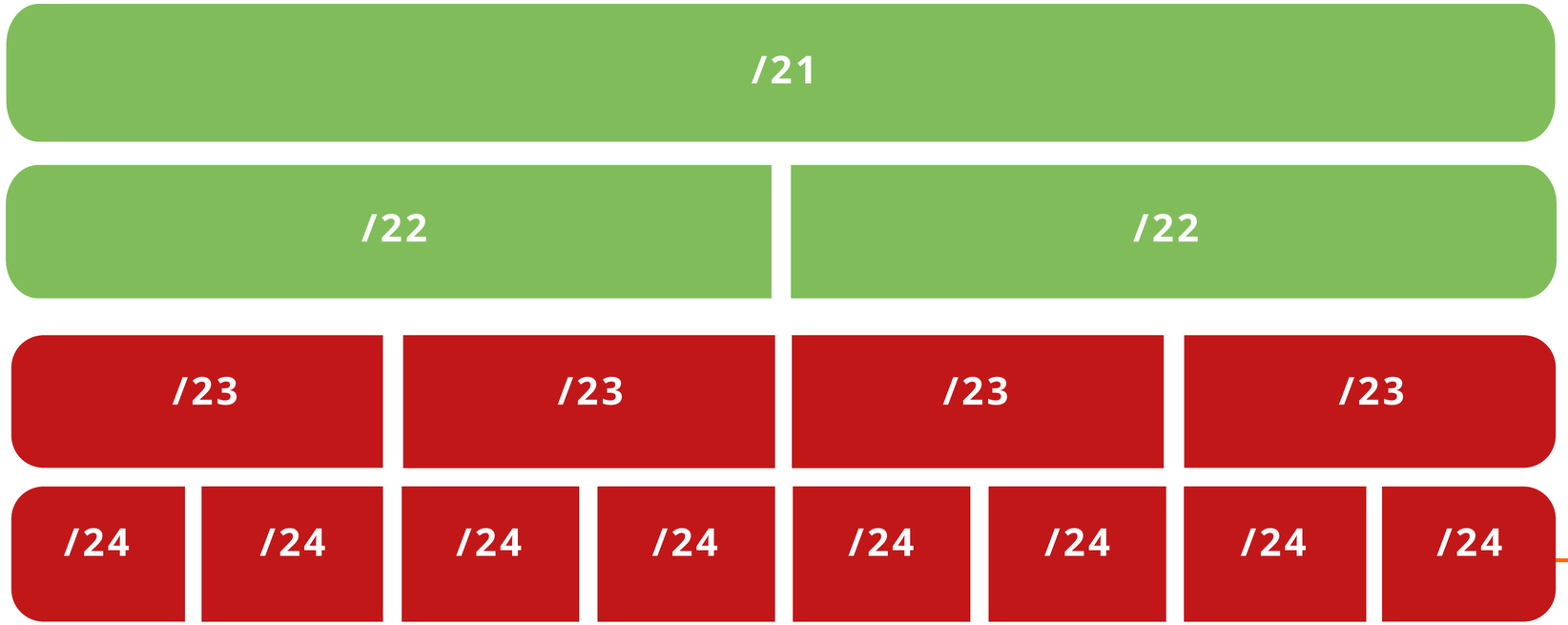
Max-Length

- RIPE NCC (AS3333) has an IP address allocation
- RIPE NCC creates this ROA
- According to the ROA:

193.0.0.0/21

ROA

Prefix	193.0.0.0/21
Max Length	/22
Origin ASN	AS3333



Any other specific announcements are unauthorised by the ROA

Questions

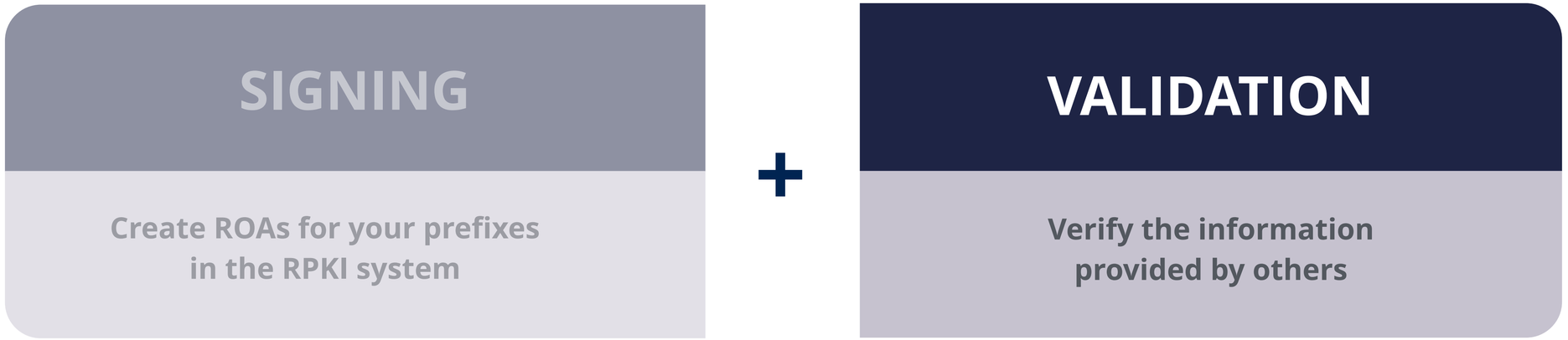
Demo

- RIPE NCC RPKI Dashboard walkthrough
- ROA Creation in the Dashboard

RPKI Validation

Elements of RPKI

- The RPKI system consists of two parts:



RPKI Validation

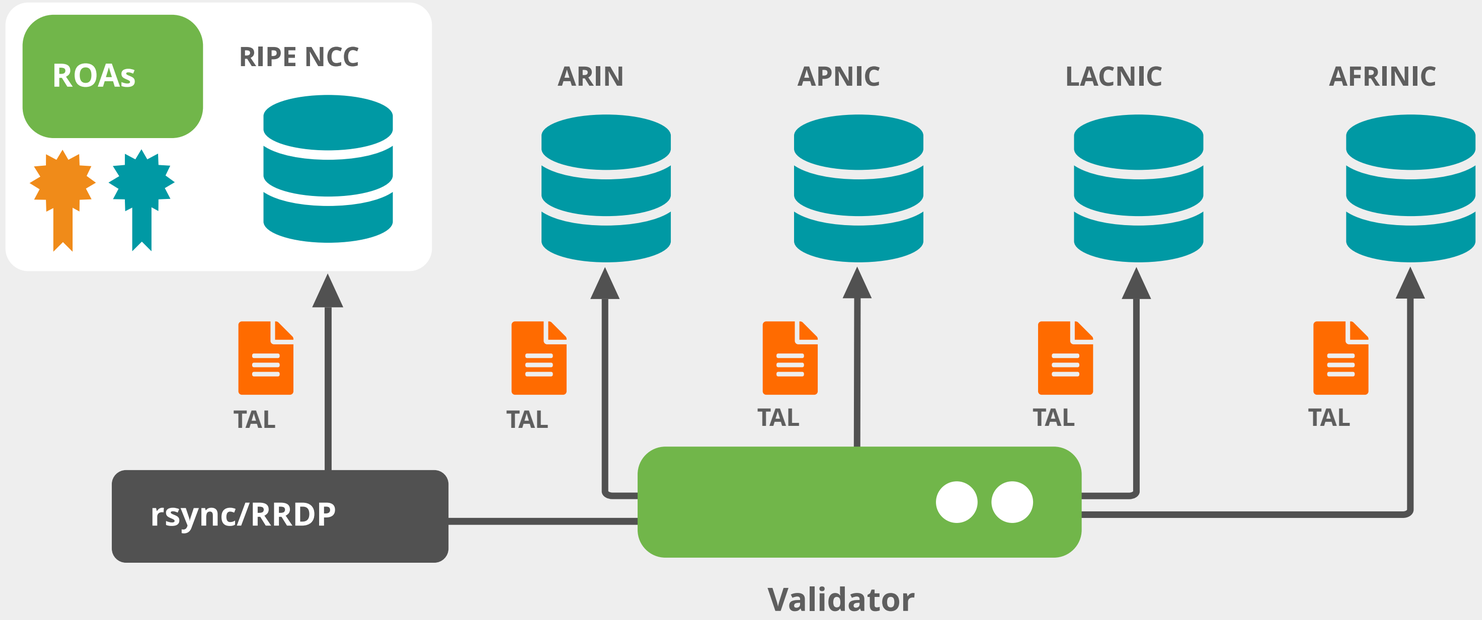
- Verifying the information provided by others

- First, **validate the RPKI data**
  - Install a validator software locally in your network
  - Verify holdership through a public key and certificate infrastructure

- Second, **validate the origin** of BGP announcements
  - Known as BGP Origin Validation (**BGP OV**) or Route Origin Validation (**ROV**)
  - This is done in a BGP router in your network

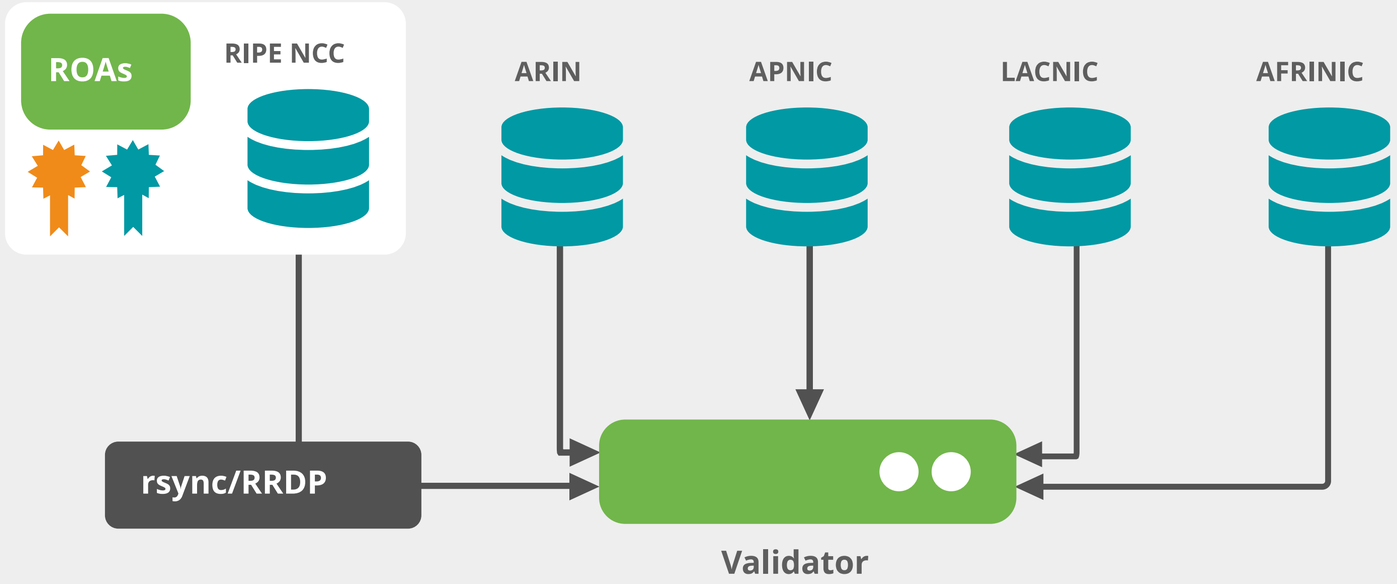
RPKI Validation

- Also known as **Relying Party (RP)** software
- Connects to RPKI repositories via **rsync** or **RRDP** protocol
- Uses information in TALs to connect to the repositories



RPKI Validation

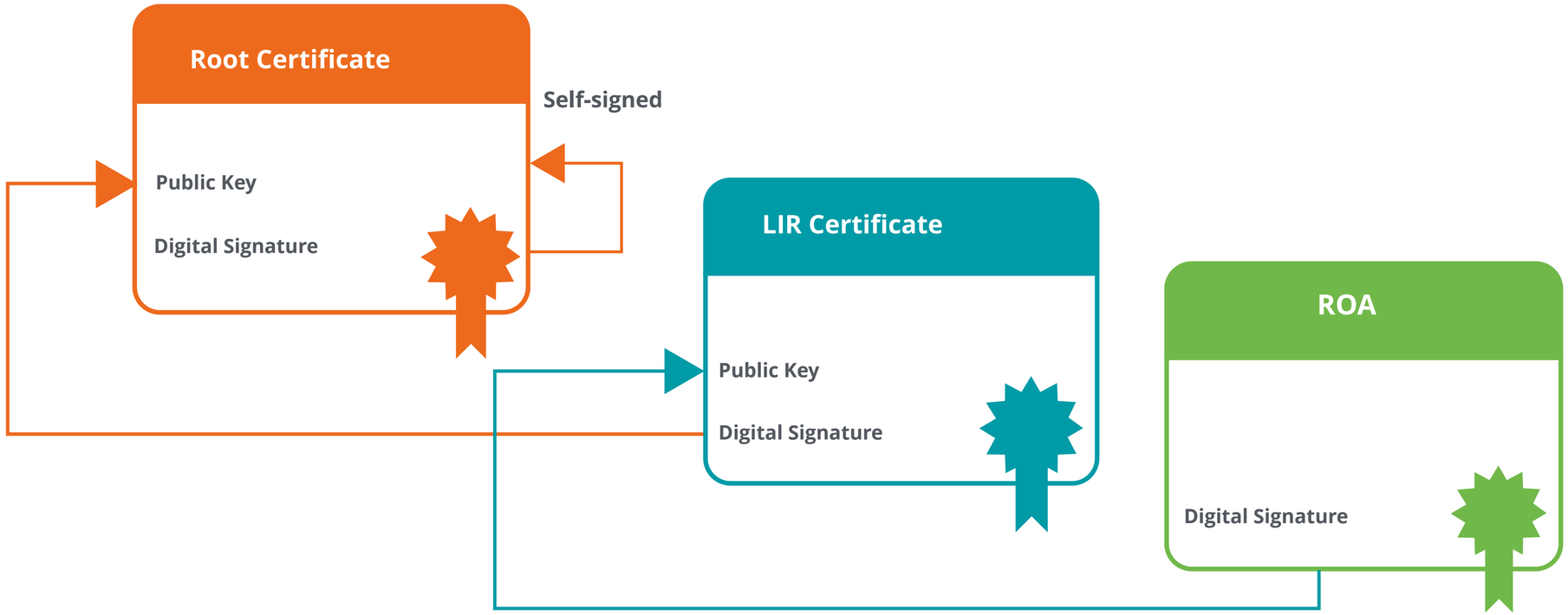
- Downloads ROAs from RPKI repositories
  - From RIRs and external repos
- Validates the chain of trust for all ROAs and associated CAs
  - Creates a local “validated cache” with all the valid ROAs



ROA Validation Process

IF chain is complete ==> ROA is **VALID!**

ELSE validation is unsuccessful ==> ROA is **INVALID!**



RPKI Validator Options

- **Routinator**
  - Built by NLNetlabs
- **rpki-client**
  - Integrated in OpenBsd
- **FORT**
  - Open source RPKI validator

Links for RPKI Validators:

https://github.com/NLnetLabs/routinator.git

https://github.com/NICMx/FORT-validator/

https://www.rpki-client.org/

More Information:

https://rpki.readthedocs.io

Which one to chose?

- You need to evaluate the different available relying party softwares
- Check HW/SW requirements
- Check dev language
- Check whether it is maintained or not
- Check whether there is an active community behind it
- Potential support options

Which one to chose?

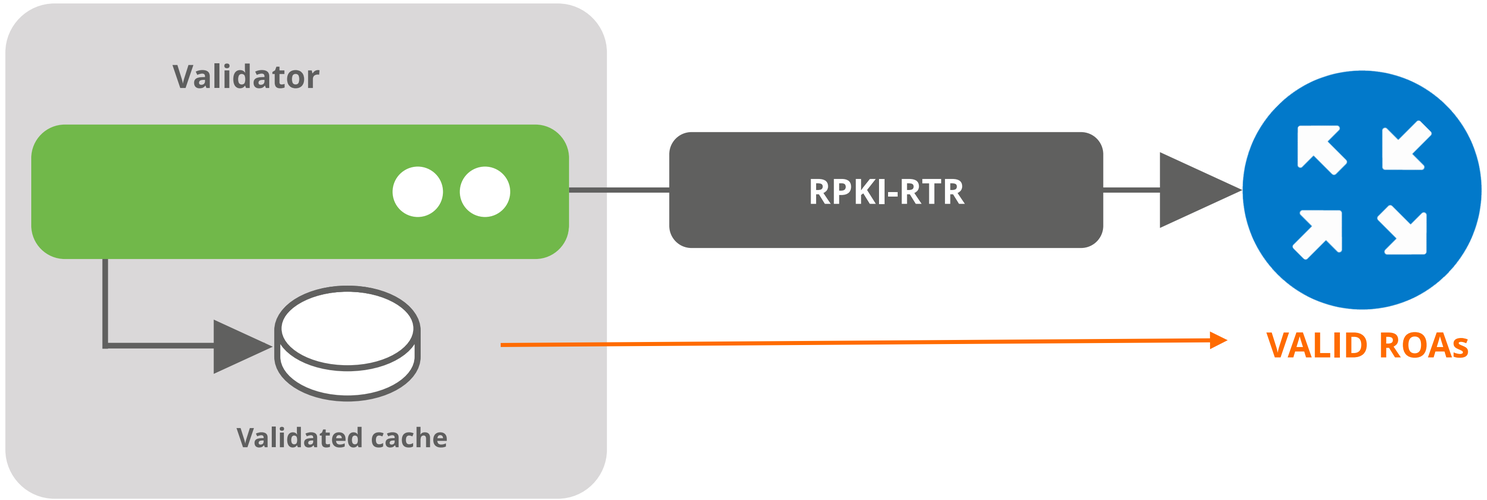
Name	Maintainer	Language	Last Commit
FORT Validator	NIC.mx	C	yesterday
OctoRPKI	Cloudflare	Go	february
rcynic	Dragon Research Labs	Python 2	december 2021
Routinator	NLnet Labs	Rust	may
rpki-client	OpenBSD	C	april
rpki-prover	Misha Puzanov	Haskell	march
RPSTIR2	ZDNS	Go	july 2023

Source: https://rpki.readthedocs.io/ 18 May 2024

Which one to chose?

- When deploying RPKI Validators, think about redundancy
- Deploy at least 2 validators
- Preferably in different locations / networks
- Preferably from different providers

Valid ROAs are sent to the router



Router uses this information to make better routing decisions

OR

RTR Server software

- In some scenarios, you could use a RTR software to proxy the RPKI Data
- RTR Server gets feed from one or multiple RPKI Validators
- Allows multiple deployment architectures
- Useful if RPKI Validator is centralised and you need distributed local RTR Caches towards the edge

RTR Server Options

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

Source: https://rpki.readthedocs.io/ 18 May 2024

Demo

- Running a validator and connecting a router to it

10 min

Start the Validator (Fort)

Initialise the validator service

```
fort --init-tals --tal=/etc/fort/tal/
```

Enable the FORT service

```
systemctl enable --now fort
```

Check if it’s running

```
ss -tlnp | grep fort
```

Check Status (Fort)

```
[root@validator ~]# ss -tlnp | grep fort
LISTEN     0      128    100.64.1.1:323                    *:*
users:(("fort",pid=1009,fd=4))
```

```
[root@validator ~]# journalctl -u fort -f
Aug 12 13:33:59 validator fort[9708]: INF: Attempting to bind socket to address
'100.64.1.1', port '323'.
Aug 12 13:33:59 validator fort[9708]: INF: Success; bound to address '100.64.1.1',
port '323'.
Aug 12 13:33:59 validator fort[9708]: WRN: First validation cycle has begun, wait
until the next notification to connect your router(s)
Aug 12 13:33:59 validator fort[9708]: INF: Starting validation.
Aug 12 13:34:00 validator fort[9708]: INF: Checking if there are new or modified
SLURM files
Aug 12 13:34:00 validator fort[9708]: INF: Applying configured SLURM
Aug 12 13:34:00 validator fort[9708]: INF: Validation finished:
Aug 12 13:34:00 validator fort[9708]: INF: - Valid ROAs: 200
Aug 12 13:34:00 validator fort[9708]: INF: - Valid Router Keys: 0
Aug 12 13:34:00 validator fort[9708]: INF: - Serial: 1
Aug 12 13:34:00 validator fort[9708]: INF: - Real execution time: 1 secs.
Aug 12 13:34:00 validator fort[9708]: WRN: First validation cycle successfully
ended, now you can connect your router(s)
<Press Ctrl+C to exit>
```

Check VRPs (Fort)

```
[root@validator ~]# grepcidr 2001:ff01::/32 /var/lib/fort/
roas.csv
AS101,2001:ff01::/32,32
```

Start the Validator (Routinator)

Enable the Routinator service

```
systemctl enable --now routinator
```

Check if it’s running

```
ps aux | grep routinator
```

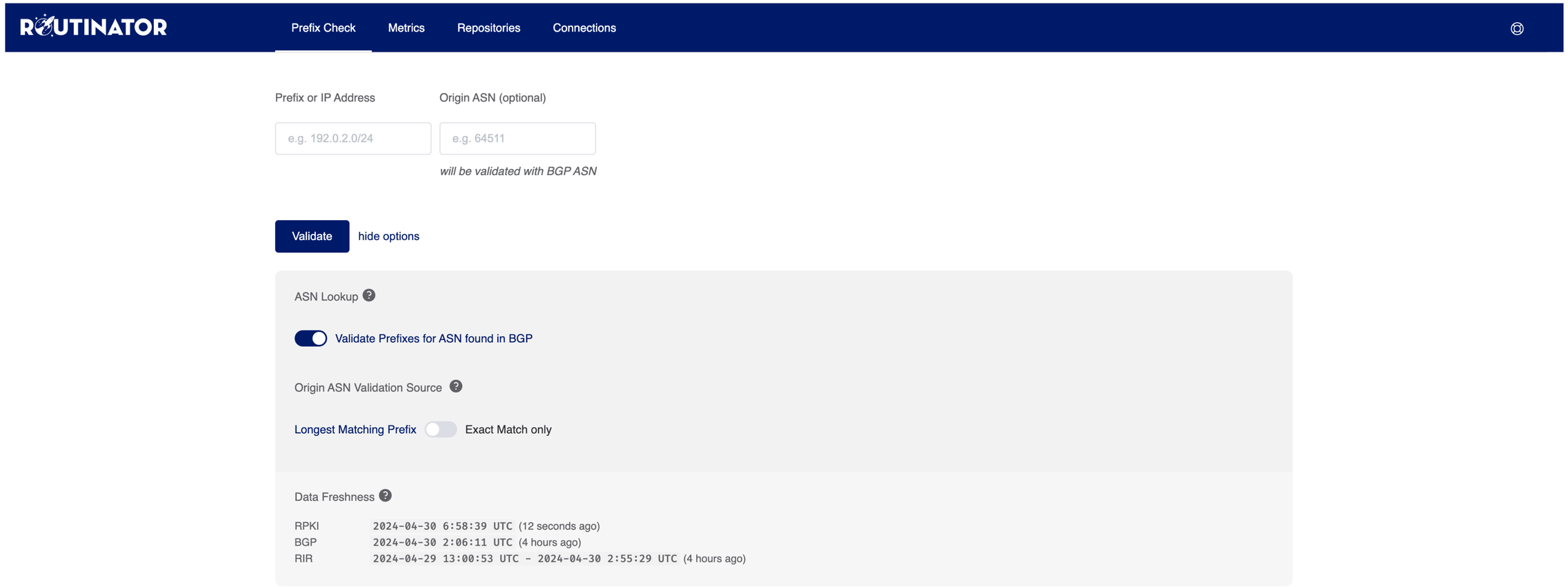
Check Status (Routinator)

```
[root@validator ~]# curl -s http://localhost:3323/status
version: routinator/0.12.1
serial: 0
last-update-start-at:  2023-01-19 12:31:04.503227799 UTC
last-update-start-ago: PT34.087042801S
last-update-done-at:   2023-01-19 12:31:05.148711439 UTC
last-update-done-ago:  PT33.441559161S
last-update-duration:  PT0.645483640S
valid-roas: 71
valid-roas-per-tal: ripe-ncc-pilot=71
vrps: 332
vrps-per-tal: ripe-ncc-pilot=332
locally-filtered-vrps: 0
locally-filtered-vrps-per-tal: ripe-ncc-pilot=0
duplicate-vrps-per-tal: ripe-ncc-pilot=0
locally-added-vrps: 0
final-vrps: 332
final-vrps-per-tal: ripe-ncc-pilot=332
stale-count: 0
```

Check VRPs (Routinator)

```
[root@validator ~]# curl -s http://localhost:3323/csv |
grepcidr 193.0.24.0/21
AS2121, 193.0.24.0/21,21,ripe-ncc-pilot
```

UI Walkthrough



Connect your router to the Validator

Example 1 - Cisco

```
(config)# conf t
(config)# router bgp 100
(config-router)# bgp rpki server tcp 100.64.1.1 port 323 refresh 300
(config-router)# bgp rpki server tcp 100.64.10.1 port 323 refresh 300
```

Check that it connected properly

Example 1 - Cisco

```
Router#show ip bgp rpki servers
BGP SOVC neighbor is 100.64.1.1/323 connected to port 323
Flags 64, Refresh time is 300, Serial number is 80, Session ID is
31990
.
.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 100.64.1.2, Local port: 31795
Foreign host: 100.64.1.1, Foreign port: 323
.
.
BGP SOVC neighbor is 100.64.1.1/3323 connected to port 323
Flags 64, Refresh time is 300, Serial number is 0, Session ID is 31627
.
.
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Mininum incoming TTL 0, Outgoing TTL 255
Local host: 100.64.1.2, Local port: 29760
Foreign host: 100.64.10.1, Foreign port: 323
```

Check received validated prefixes

Example 1 - Cisco

```
Router#sh ip bgp rpki table
90 BGP sovc network entries using 14400 bytes of memory
90 BGP sovc record entries using 2880 bytes of memory

Network            Maxlen  Origin-AS  Source  Neighbor
10.1.1.0/24        24      201        0       100.64.1.1/323
10.1.1.0/24        24      201        0       100.64.10.1/323
10.1.2.0/24        24      301        0       100.64.1.1/323
10.1.2.0/24        24      301        0       100.64.10.1/323
10.2.1.0/24        24      202        0       100.64.1.1/323
10.2.1.0/24        24      202        0       100.64.10.1/323
10.2.2.0/24        24      302        0       100.64.1.1/323
10.2.2.0/24        24      302        0       100.64.10.1/323
10.3.1.0/24        24      203        0       100.64.1.1/323
10.3.1.0/24        24      203        0       100.64.10.1/323
10.3.2.0/24        24      303        0       100.64.1.1/323
10.3.2.0/24        24      303        0       100.64.10.1/323
10.4.1.0/24        24      204        0       100.64.1.1/323
10.4.1.0/24        24      204        0       100.64.10.1/323
10.4.2.0/24        24      304        0       100.64.1.1/323
--More--
```

Connect your router to the Validator

Example 2 - FRR

```
Router# conf t
Router# rpki
Router(config-rpki)# rpki polling_period 3600
Router(config-rpki)# rpki cache 2001:db8:30:30::ff   3323 preference 1
Router(config-rpki)# end
```

Check that it connected properly

Example 2 - FRR

```
Router# show rpki cache-connection
Connected to group 1
rpki tcp cache 2001:db8:30:30::ff 3323 pref 1 (connected)
```

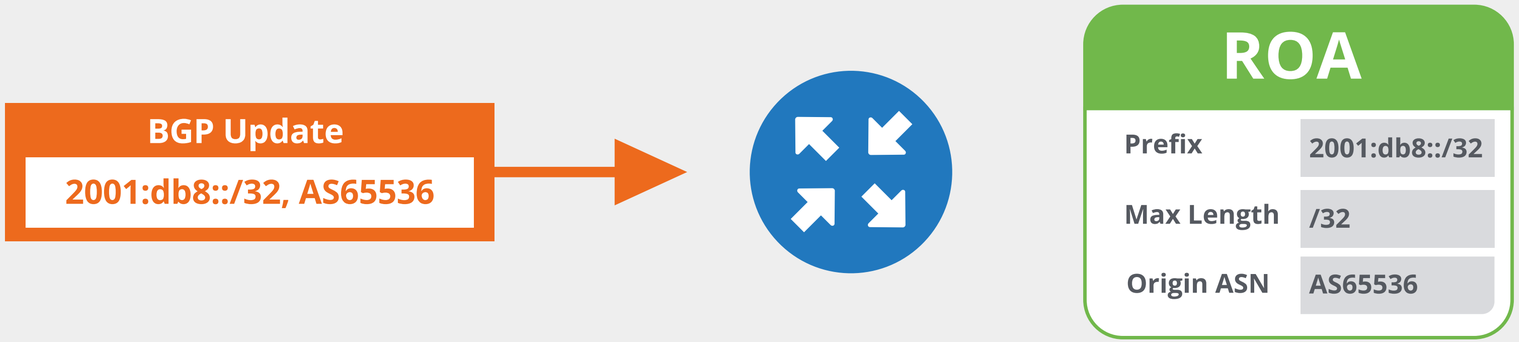
Check received validated prefixes

Example 2 - FRR

```
Router# show rpki prefix-table
RPKI/RTR prefix table
Prefix                                  Prefix Length  Origin-AS
10.3.1.0                                     24 -  24        203
10.4.1.0                                     24 -  24        204
10.1.1.0                                     24 -  24        201
...
2001:ff03::                                  32 -  32        103
2001:ff16::                                  32 -  32        116
2001:db8:412::                               48 -  48        412
2001:db8:440::                               48 -  48        440
2001:db8:460::                               48 -  48        460
2001:db8:500::                               48 -  48        500
2001:db8:540::                               48 -  48        540
2001:db8:550::                               48 -  48        550
2001:db8:560::
...
Number of IPv4 Prefixes: 90
Number of IPv6 Prefixes: 259
```

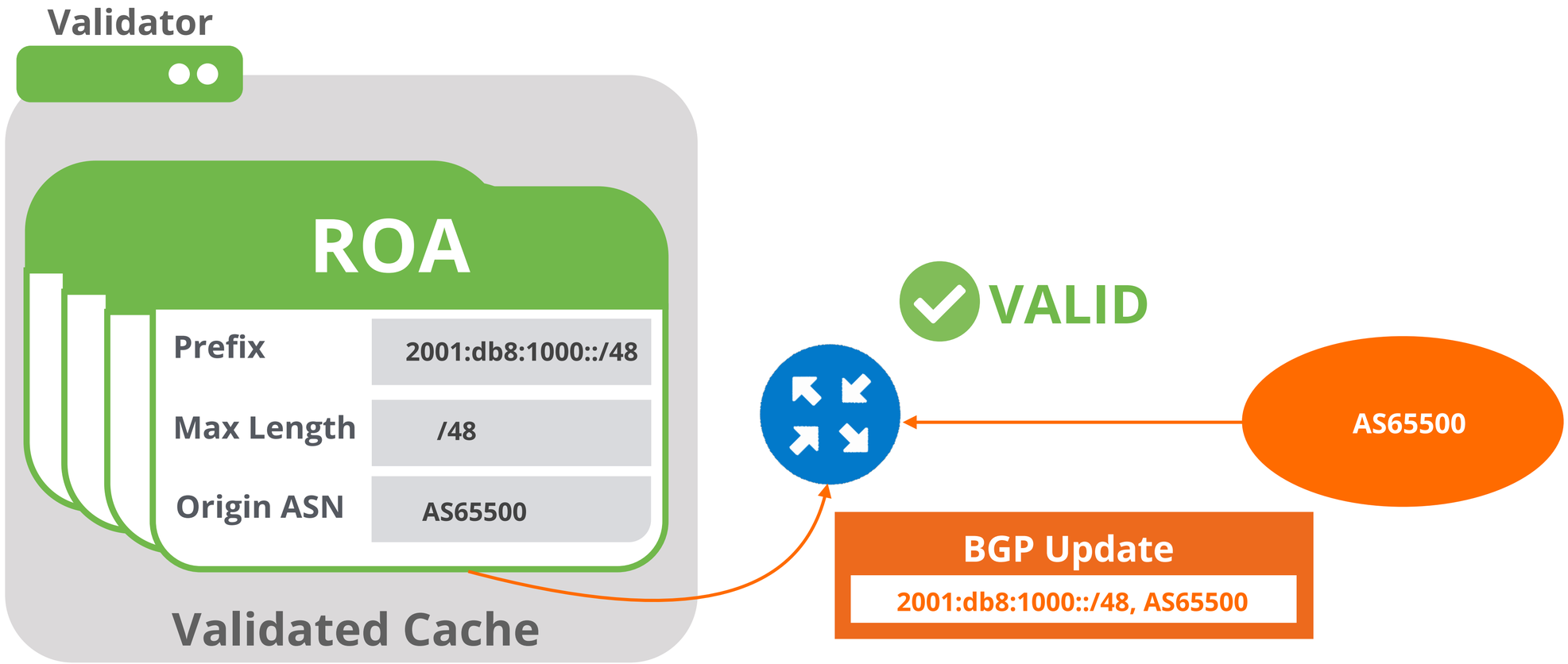
BGP Origin Validation (BGP OV)

- RPKI based route filtering
- BGP announcements are compared against the **valid** ROAs
  - **Origin ASN** and **max-length** must match
- Router decides the validation states of **routes**:
  - **Valid**, **Invalid** or **Not-Found**



RFC 6811 - BGP Prefix Origin Validation https://datatracker.ietf.org/doc/html/rfc6811

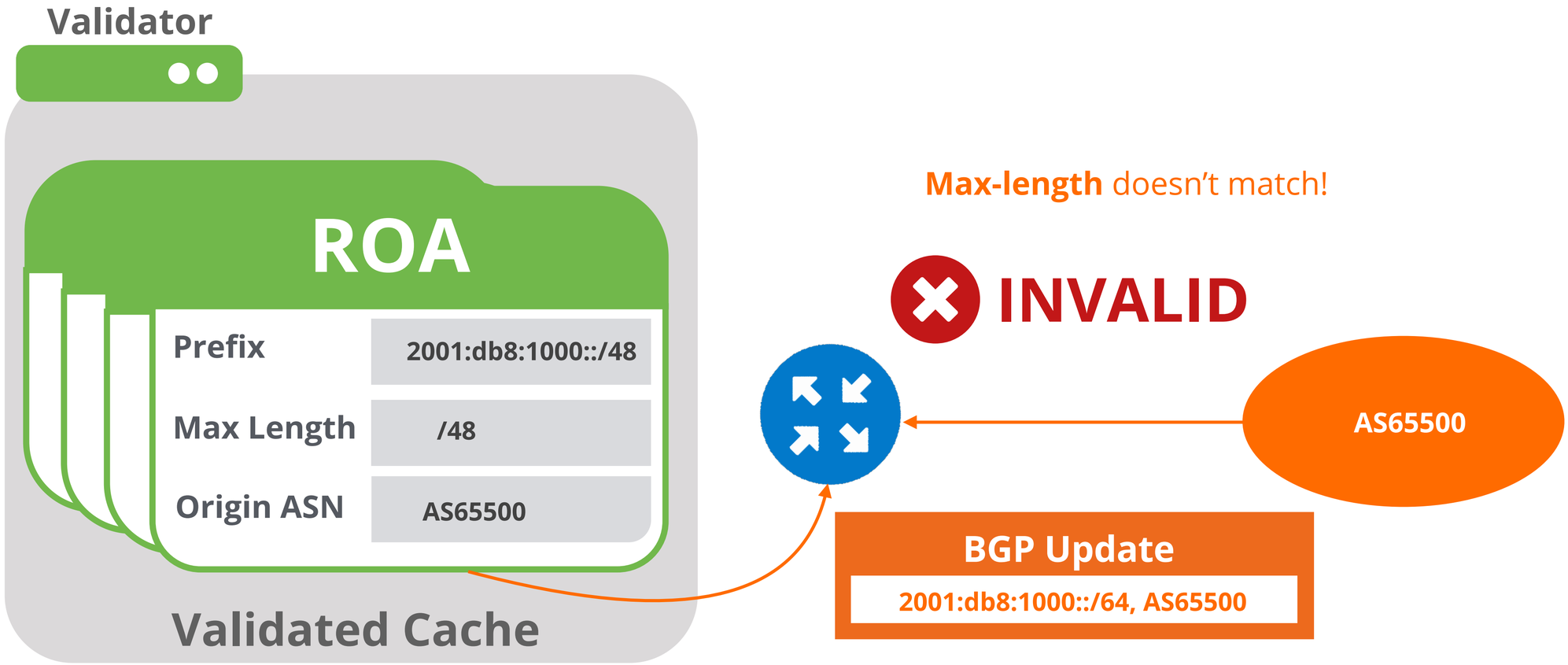
How Does RPKI Validate the Origin?



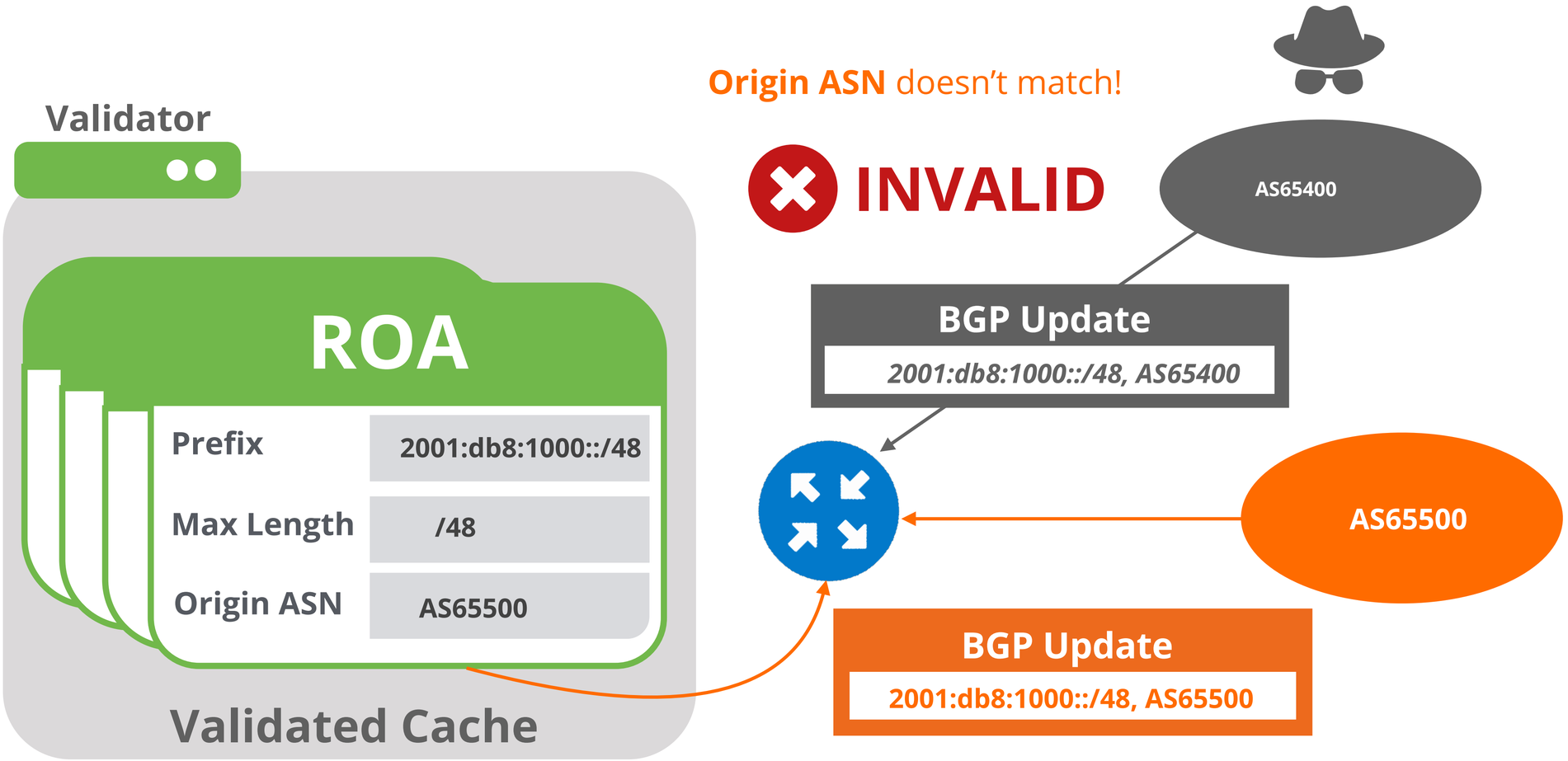
RPKI Valid

```
Router#show  ip bgp 193.0.25.0/24
BGP routing table entry for 193.0.25.0/24, version 1598443
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  99 102
    192.168.1.2 from 192.168.1.254 (99.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      path 7FD8EAB30678 RPKI State valid
      rx pathid: 0, tx pathid: 0x0
```

How Does RPKI Validate the Origin?



How Does RPKI Validate the Origin?

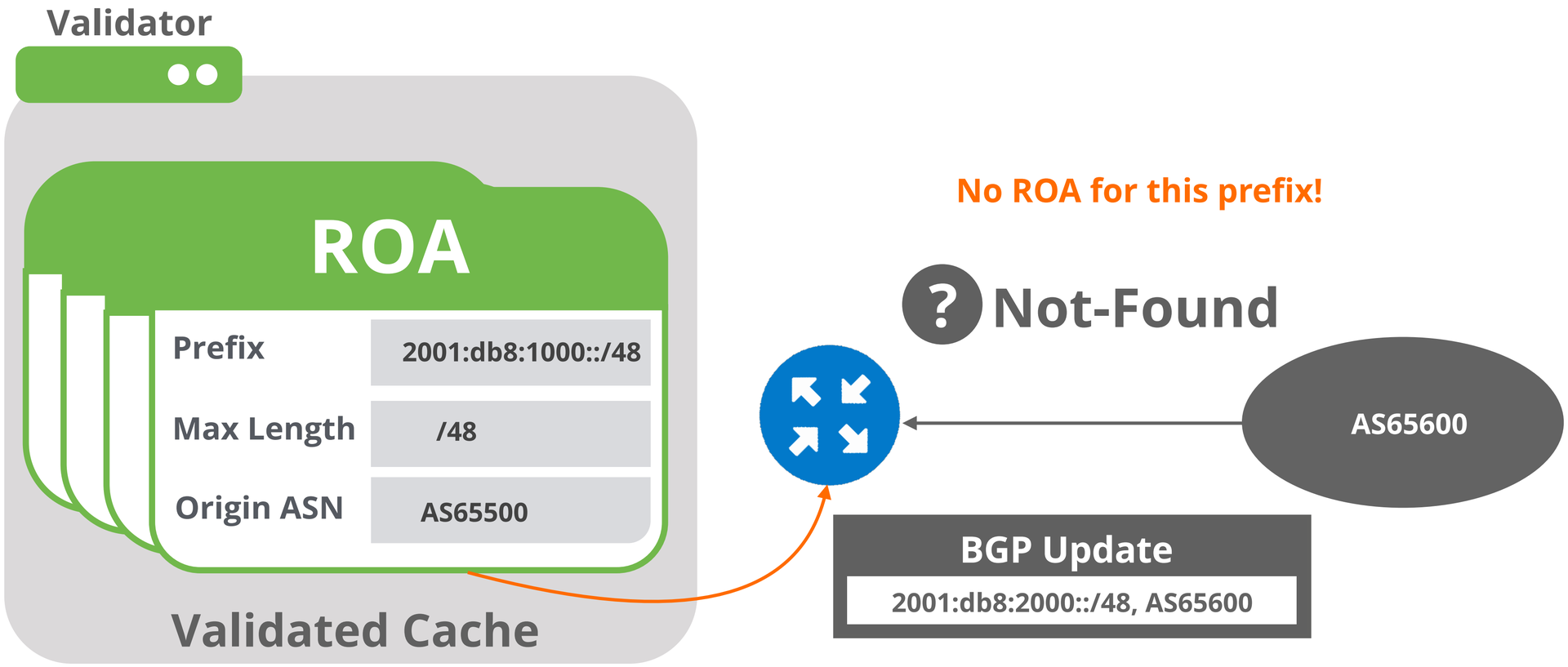


RPKI Invalid

Prefix belongs to another AS!

```
Router#show  ip bgp 193.0.26.0/24
BGP routing table entry for 193.0.26.0/24, version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 1
  99 102
    192.168.1.2 from 192.168.1.254 (99.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, external
      path 7FD8EAB30708 RPKI State invalid
      rx pathid: 0, tx pathid: 0
```

How does RPKI Validate the Origin?



Prefix Without a ROA

No ROA for this one!

```
Router#show  ip bgp 20.20.20.0/24
BGP routing table entry for 20.20.20.0/24, version 1598444
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  99 102
    192.168.1.2 from 192.168.1.254 (99.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      path 7FD8EAB305E8 RPKI State not found
      rx pathid: 0, tx pathid: 0x0
```

The General Rule

IF ROA exists that validates the prefix

↓

The prefix is **Valid**

ELSE IF any ROA invalidates the prefix

↓

The prefix is **Invalid**

↓

ELSE

The prefix I **not-found**

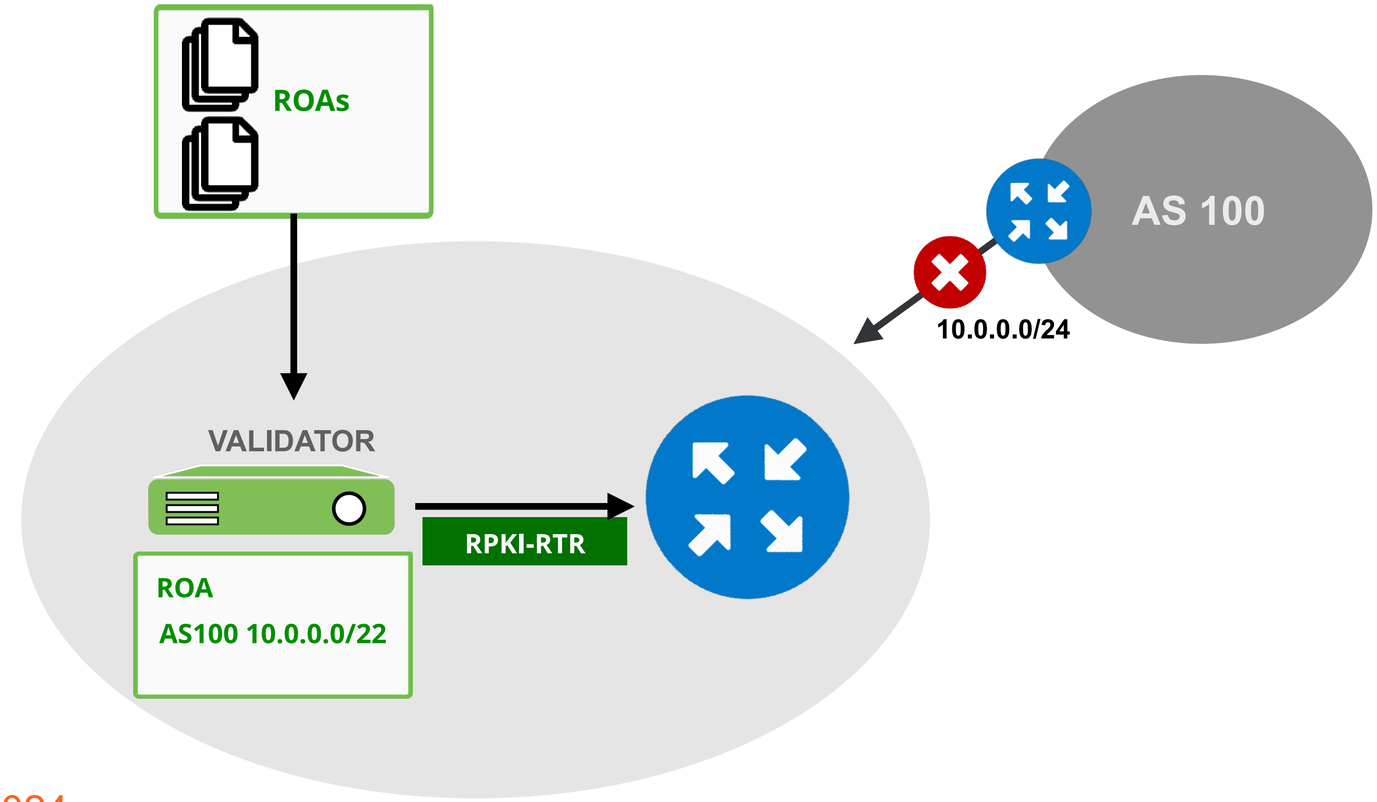
Demo

- Check Validation Status

10 min

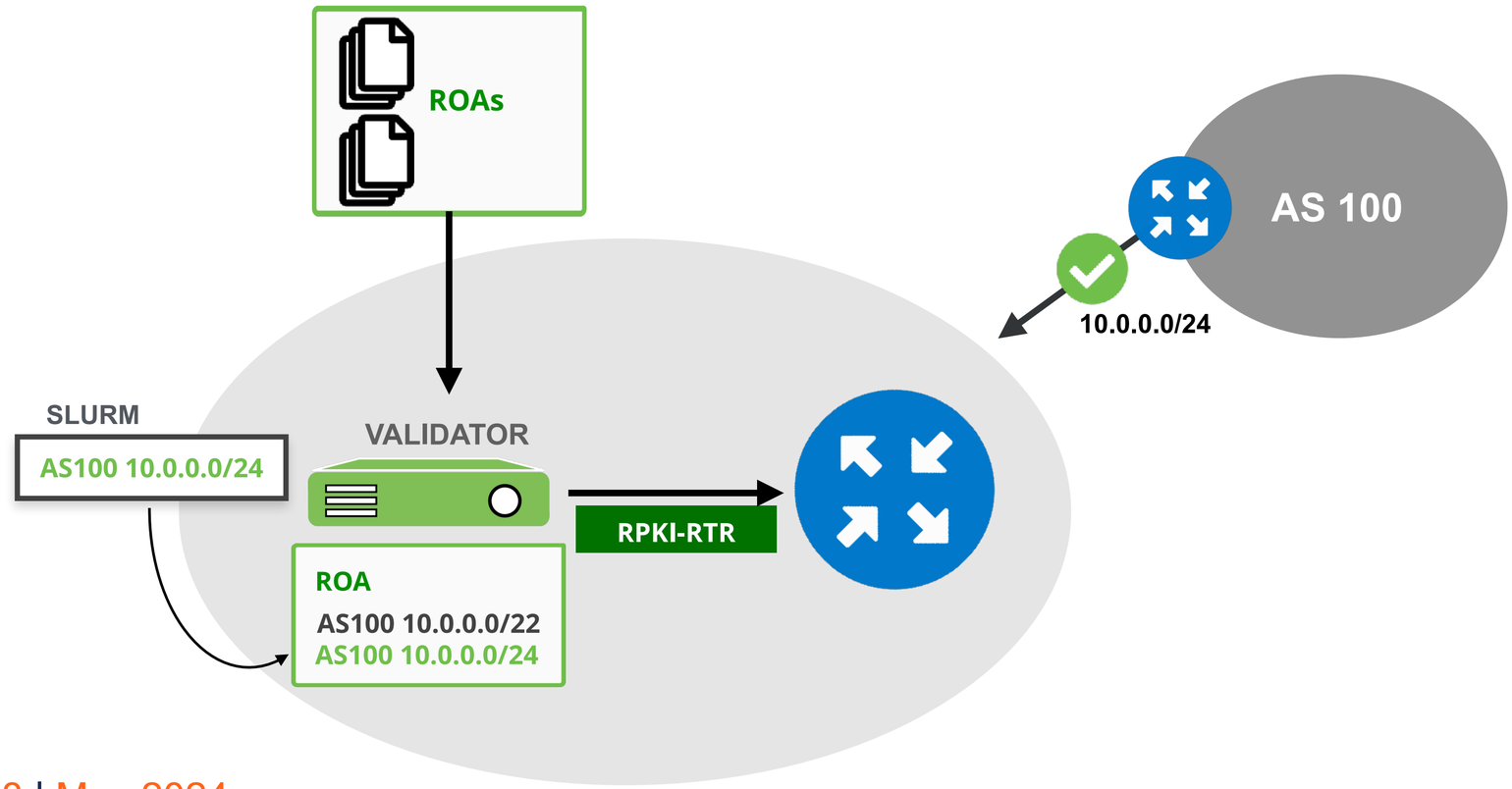
Local Overrides

- Sometimes, there is an operational need to accept invalid BGP announcements



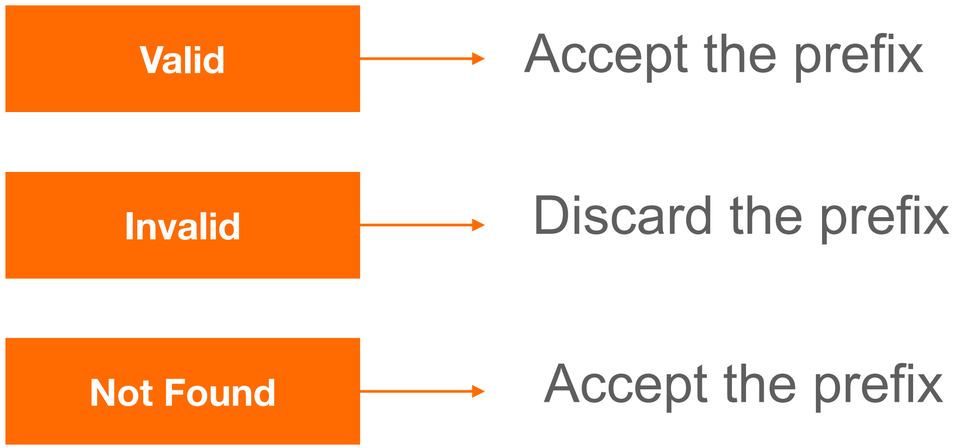
Local Overrides

- SLURM (RFC 8416) is used to temporarily change the validation status
- Use with care



After Validating

- You have to make a decision: Accept or Discard



Do not consider dropping prefixes with “Not-Found” RPKI validation state!

Configure Validation Policy

Configure Route-map on your BGP Router

```
(config-router)# route-map rpki-accept permit 10
(route-map)# match rpki valid
(route-map)# set local-preference 110

(route-map)# route-map rpki-accept permit 20
(route-map)# match rpki not-found
(route-map)# set local-preference 80
```

Add Route Map to Neighbour

```
(config)# router bgp 101
(config)# address-family ipv4
(config)# neighbor 192.168.1.254 route-map rpki-accept in
```

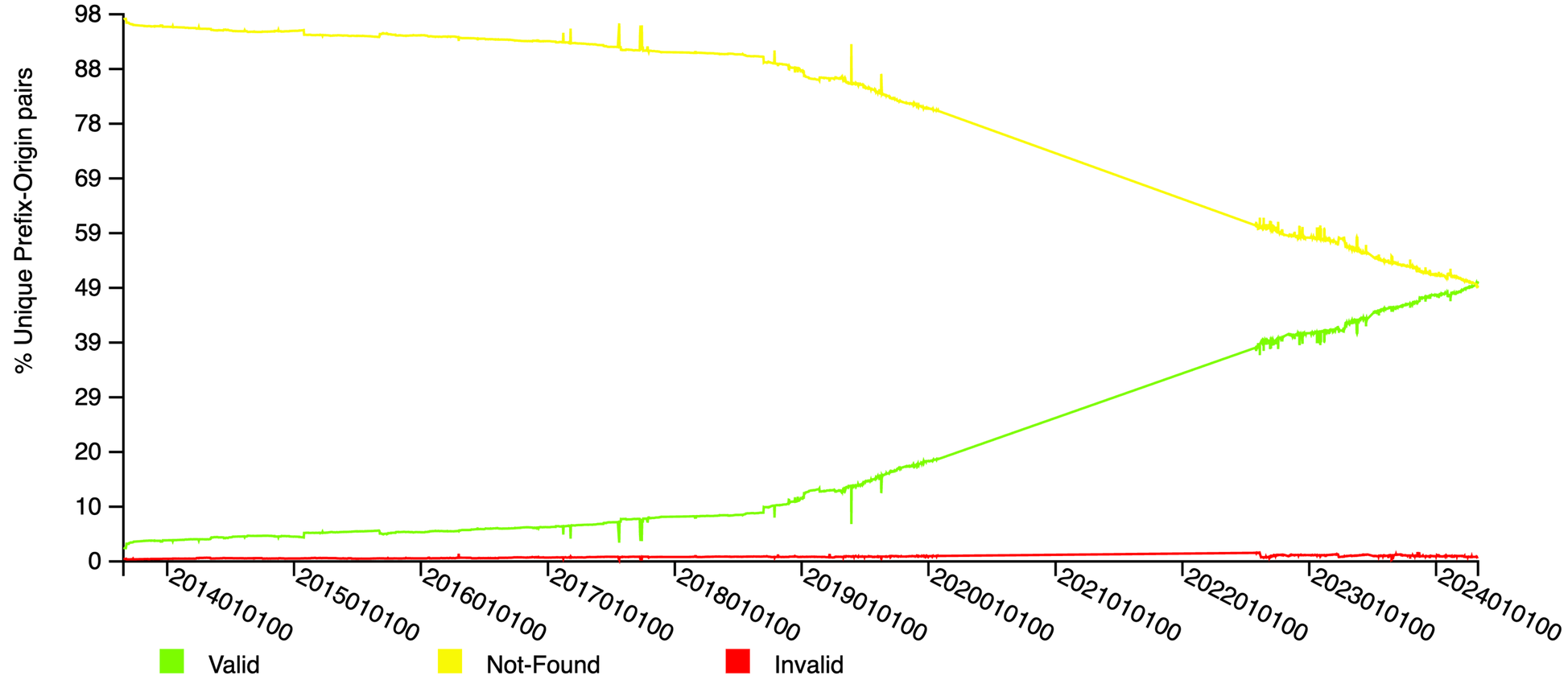
Major Networks and RPKI Invalids

- Major networks are dropping invalids
  - Telia, AT&T, Cloudflare, Netflix, Swisscom, Cogent etc

- They follow a phased approach: First peers, then customers
  - Tag invalids on all peers, then on all customers
  - Drop invalids for all peers, then for all customers

For more information: https://isbgpsafeyet.com/

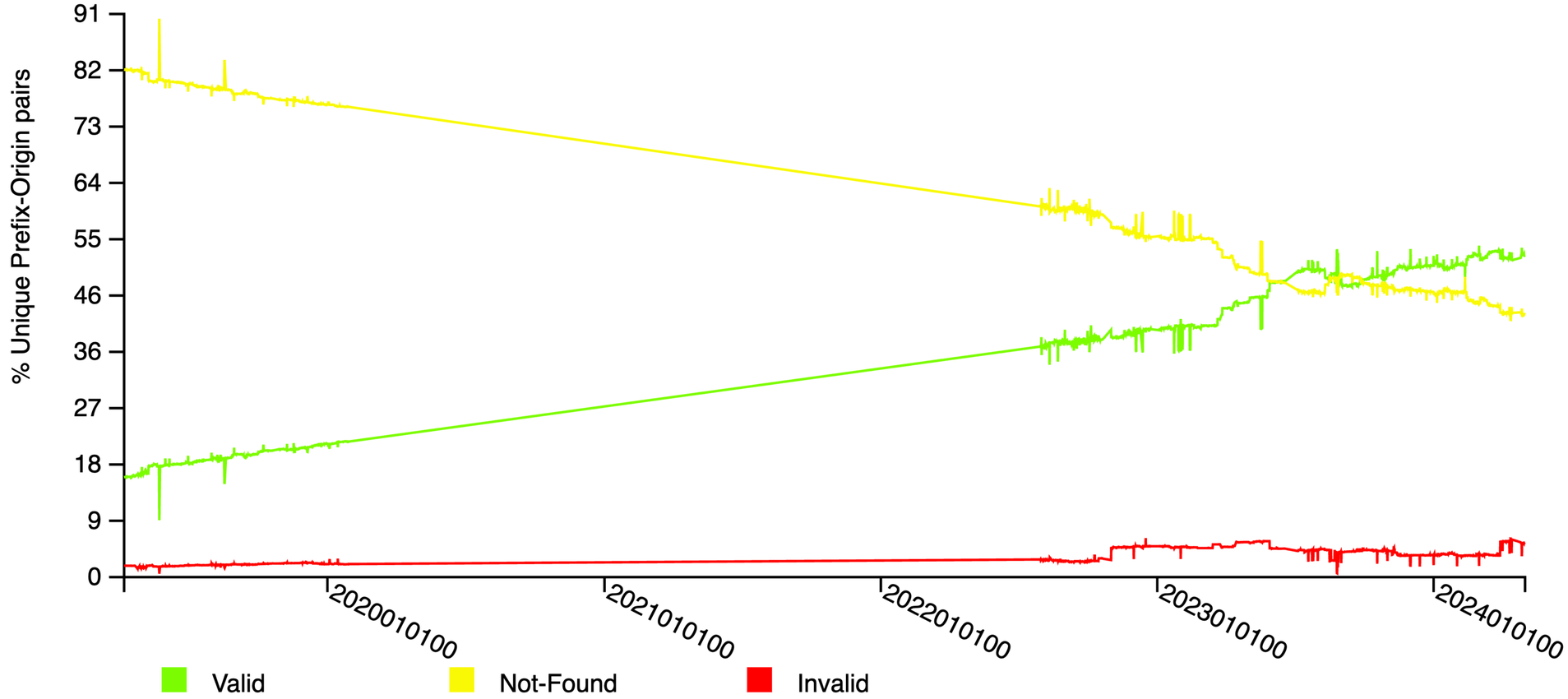
RPKI-ROV Analysis Globally (IPv4)



NIST RPKI Monitor: RPKI-ROV Analysis **Protocol:** IPv4 **RIR:** All

More information (IPv4, IPv6, By RIR, Date): https://rpki-monitor.antd.nist.gov

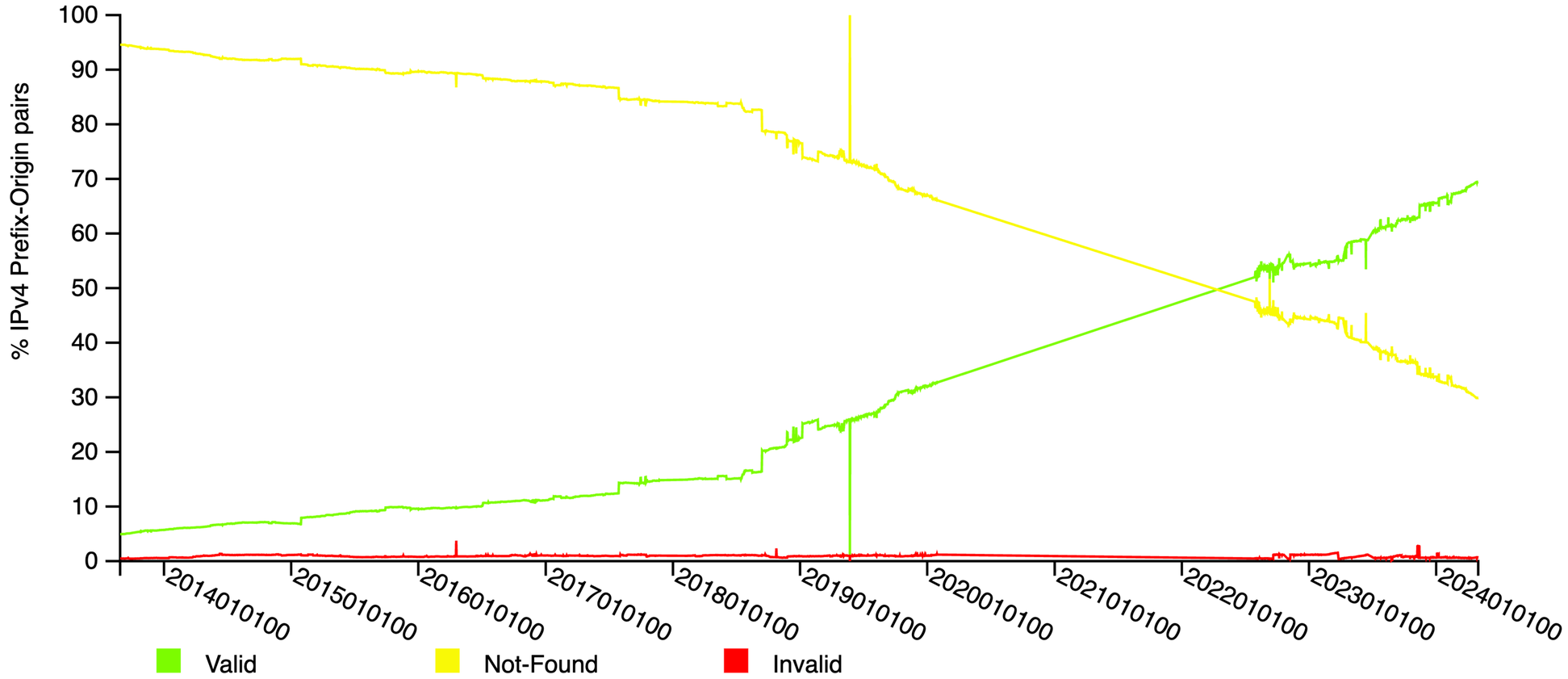
RPKI-ROV Analysis Globally (IPv6)



NIST RPKI Monitor: RPKI-ROV Analysis **Protocol:** IPv6 **RIR:** All

More information (IPv4, IPv6, By RIR, Date): https://rpki-monitor.antd.nist.gov

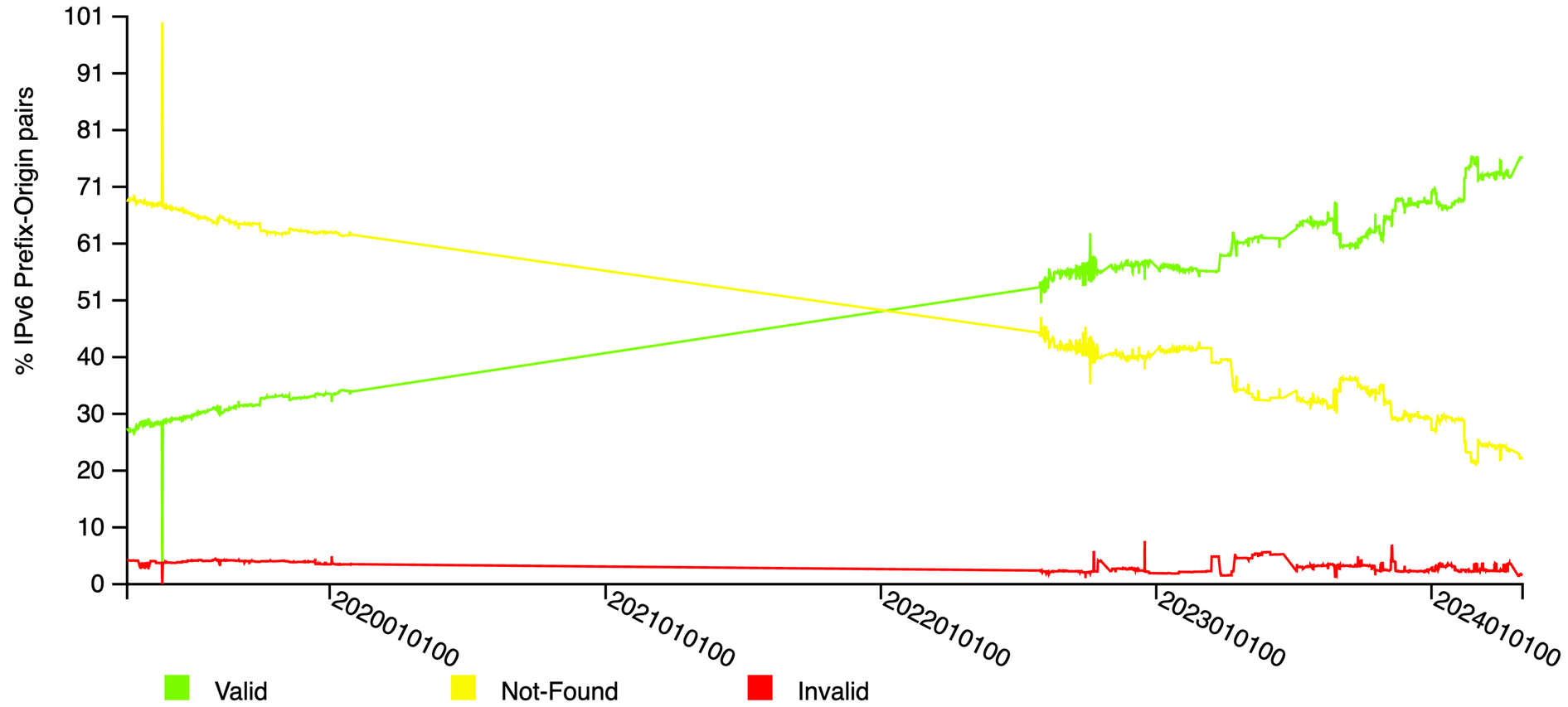
RPKI-ROV Analysis RIPE Region (IPv4)



NIST RPKI Monitor: RPKI-ROV Analysis **Protocol:** IPv4 **RIR:** RIPE

More information (IPv4, IPv6, By RIR, Date): https://rpki-monitor.antd.nist.gov

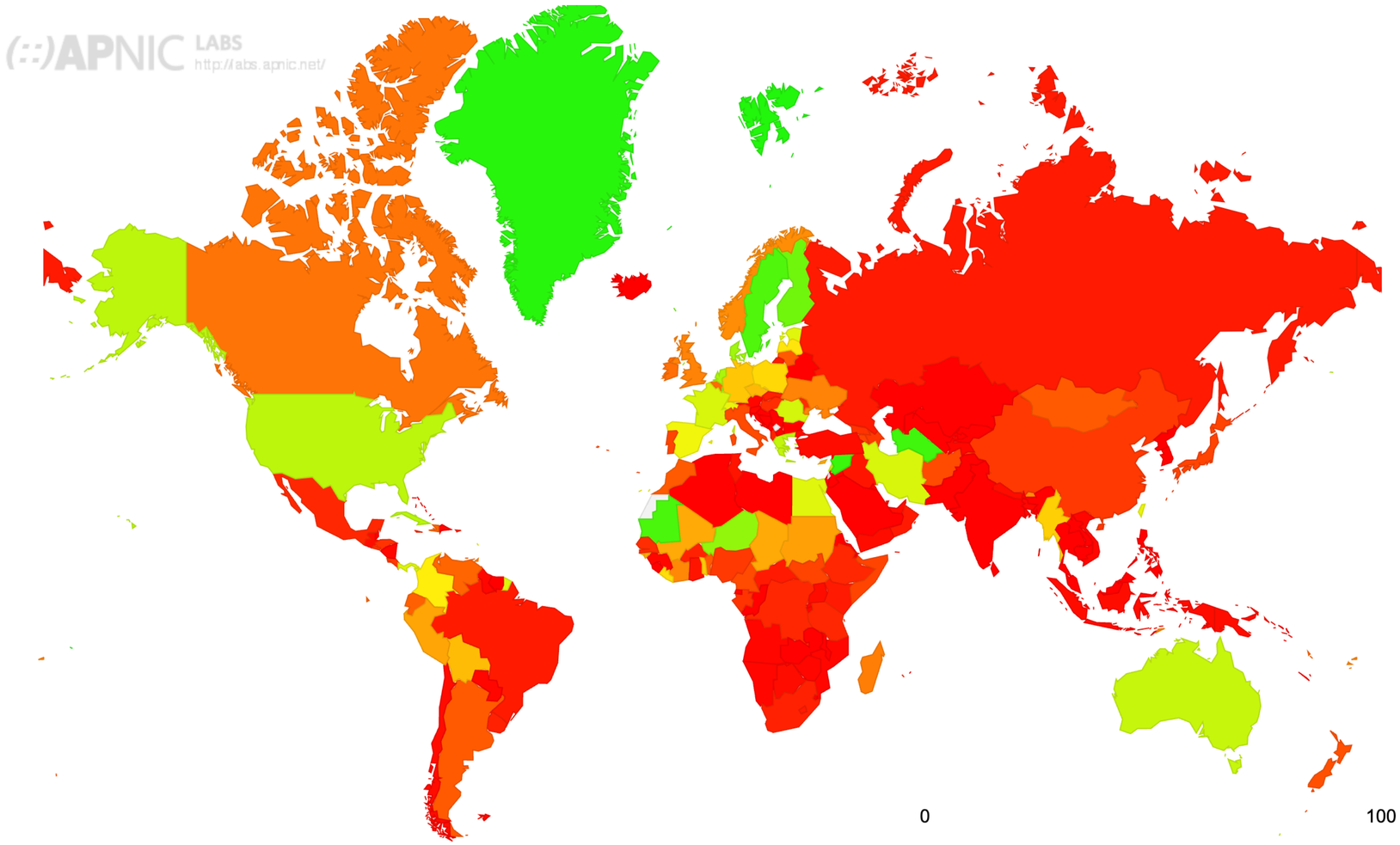
RPKI-ROV Analysis RIPE Region (IPv6)



NIST RPKI Monitor: RPKI-ROV Analysis **Protocol:** IPv6 **RIR:** RIPE

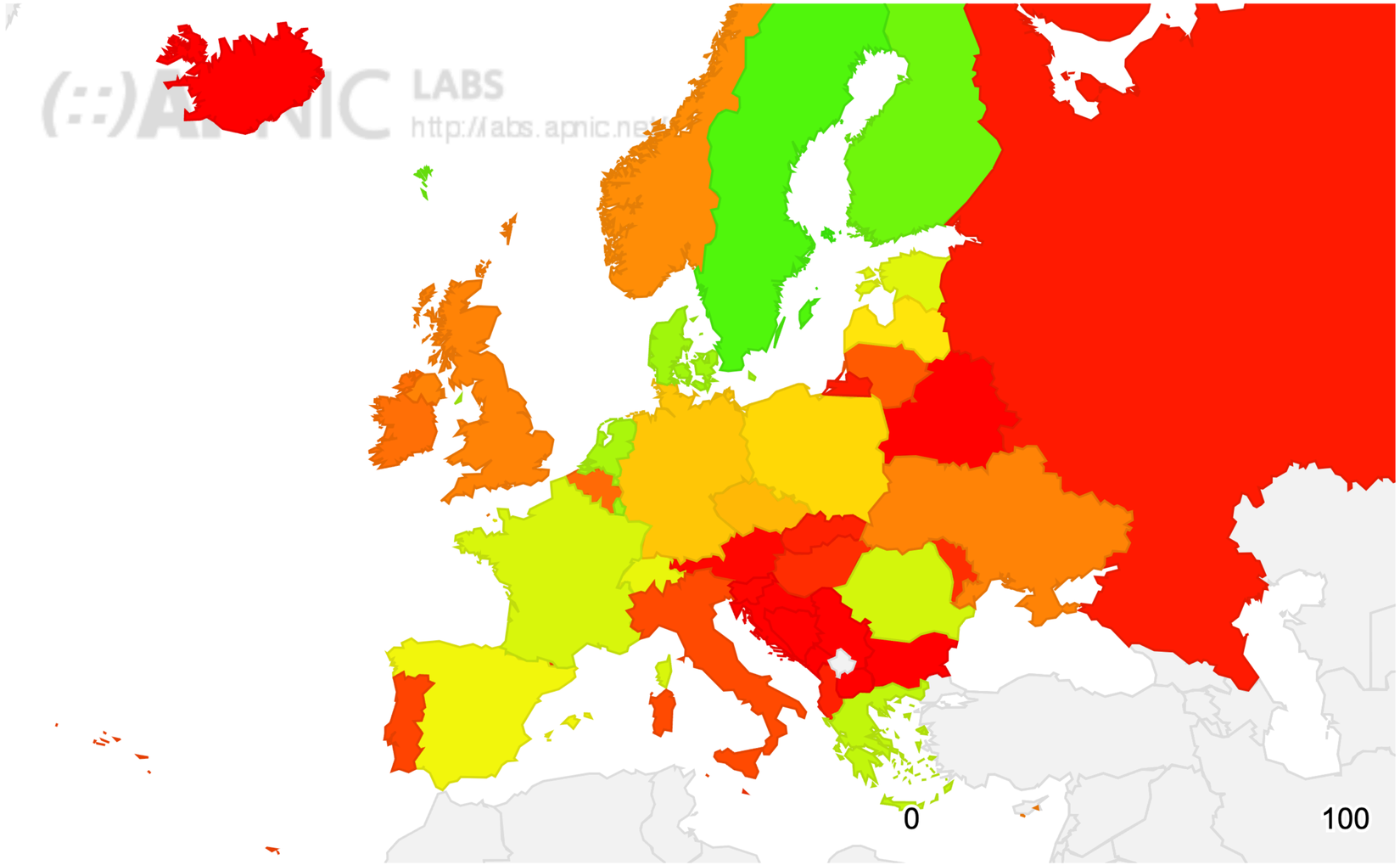
More information (IPv4, IPv6, By RIR, Date): https://rpki-monitor.antd.nist.gov

Are networks filtering based on RPKI data - Global?



Source: APNIC

Are networks filtering based on RPKI data - Europe?



Source: APNIC

A global RPKI ecosystem enhances routing security!

- RPKI is a powerful mechanism
  - Prevents BGP hijacks, mis-originations and route leaks
  - Currently used for validating the origin AS
  - Stepping stone to Full BGP path validation

- RPKI is opt-in
  - It will only work if every network agrees to abide by it

Let’s deploy RPKI today!

Give support for secure Internet routing
and
help to mitigate routing incidents globally!

Questions

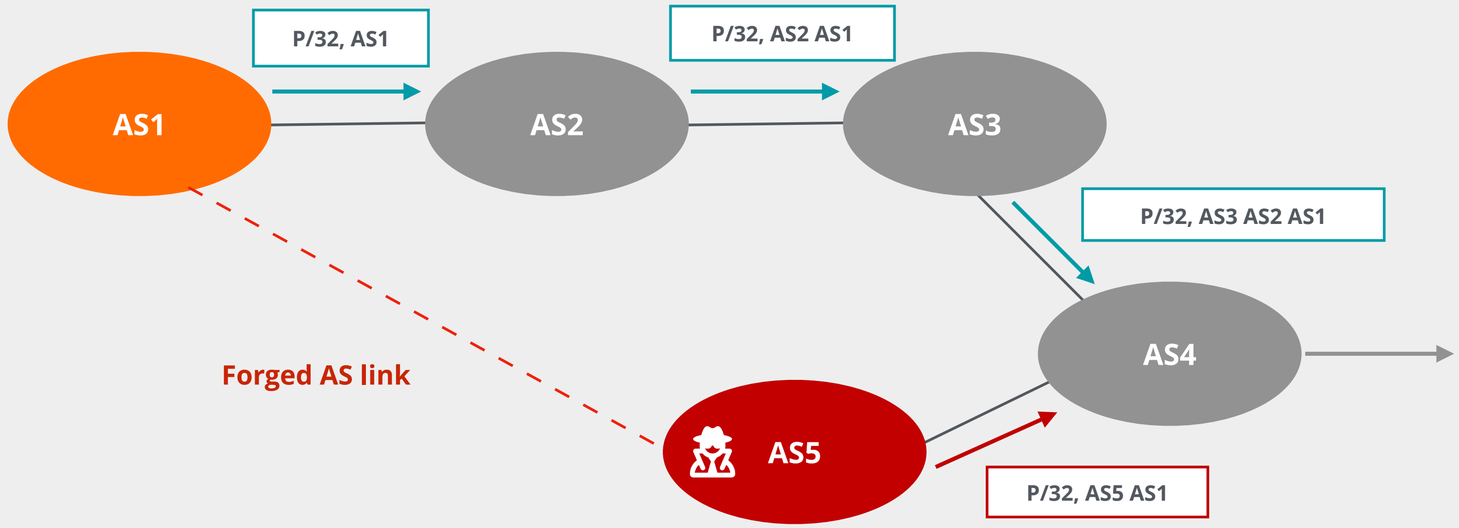
Next Steps for BGP Routing Security

What’s Next for Routing Security?

Dealing with Path Hijacks…

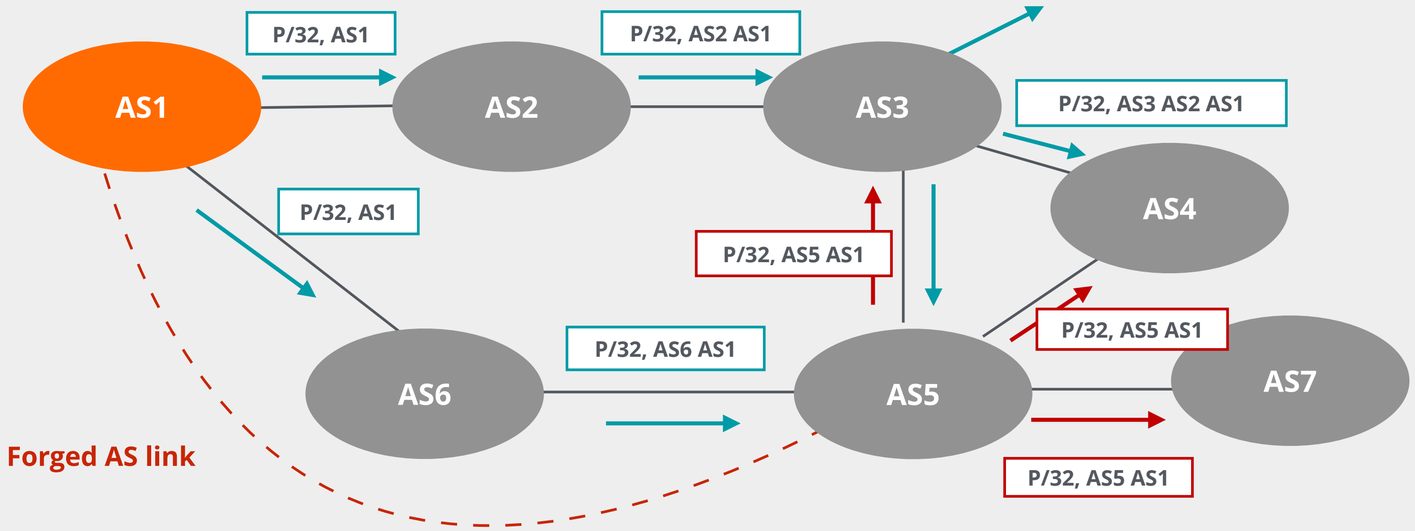
Fake Path with Correct Origin

- The origin of the path does not change!
- The attacker:
  - Creates a forged AS link between two ASes
  - Reroutes the traffic to itself



Modifying an Existing Path

- Neighbours of the attacker receive a false path
- The attacker can do either of these two things:
  - Analyse the traffic and then route to AS1
  - Drop the traffic to AS1



What's Next for Routing Security?

- RPKI today focuses on **Origin Validation!**
- But RPKI OV can not detect path manipulations!
  - Origin AS remains intact in the altered AS Path
- Then, what to do?
  - The solution is to **validate the full BGP path**
  - **Tentative solutions: BGPsec [RFC 8205] and ASPA**

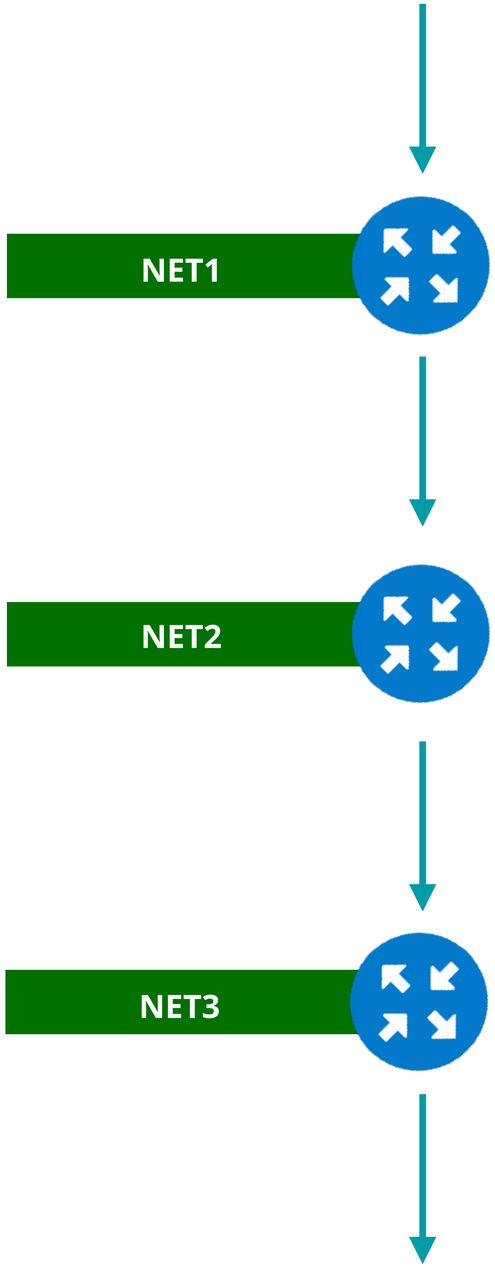
RPKI is a stepping stone to **Path validation**!

BGPSec

- Designed to supplement BGP Origin Validation
- Relies on the RPKI certificates
  - Router certificates are issued to routers within an AS
- Introduces a new BGP path attribute, **BGPsec_PATH**
  - Optional, non-transitive attribute
  - Carries digitally signed AS path information
  - Support is negotiated between BGP speakers

BGP Operations

- Routers sign the AS path in a BGP UPDATE message
- Each BGP UPDATE containing BGPsec_PATH attribute:
  - Can **advertise** a single prefix only
  - Can only be **sent** to one AS at a time
- Routers verify the chain of trust of **all of the signatures** of the AS Path



Network: 192.168.0.0/16
AS Path: NET1, ...
BGPSEC: (key1, signature1)

Network: 192.168.0.0/16
AS Path: NET2, NET1, ...
BGPSEC: (key1, signature1)
(key2, signature2)

Network: 192.168.0.0/16
AS Path: NET3, NET2, NET1, ...
BGPSEC: (key1, signature1)
(key2, signature2)
(key3, signature3)

BGPSec Has Some Limitations…

- Does not offer origin validation
- Does not prevent route leaks
- Expensive to run, requires more powerful routers
  - UPDATE messages are larger because of digital signatures
  - One UPDATE message is required for each prefix
  - BGP speakers need to perform cryptographic functions
- Does not support incremental deployment

That’s why progress is very slow and no deployment yet!

ASPA

- **A**utonomous **S**ystem **P**rovider **A**uthorisation
- Introduces a new digitally signed object, an **ASPA**
  - ASPA object defines upstreams for a defined AS
- ASPA proposes a lightweight solution for path validation
  - Leverages existing RPKI infrastructure
  - Does not require a new BGP attribute
  - Requires a database where ASPA objects could be queried
  - Verifies the sequence of ASes along the path

How Does ASPA Work?

- Customer AS creates an ASPA object and signs it
  - Authorises a set of **Provider ASes** to propagate its route announcements
- In the Validation process, receiving AS

1. Verifies that if there is a cryptographically valid ASPA for that customer
   - Is provider AS authorised to propagate a given customer's route?
2. Verifies the AS path
   - Have routes been received from a customer, a provider, or from a route server?

More About ASPA

- ASPA helps to detect route leaks and hijacks
- Incremental deployment is possible
- Still in draft state (about to become an RFC)
- Already supported in a couple of validators
- Support in OpenBGPD and NIST BGP-SRx

Questions

RIPE NCC
ACADEMY

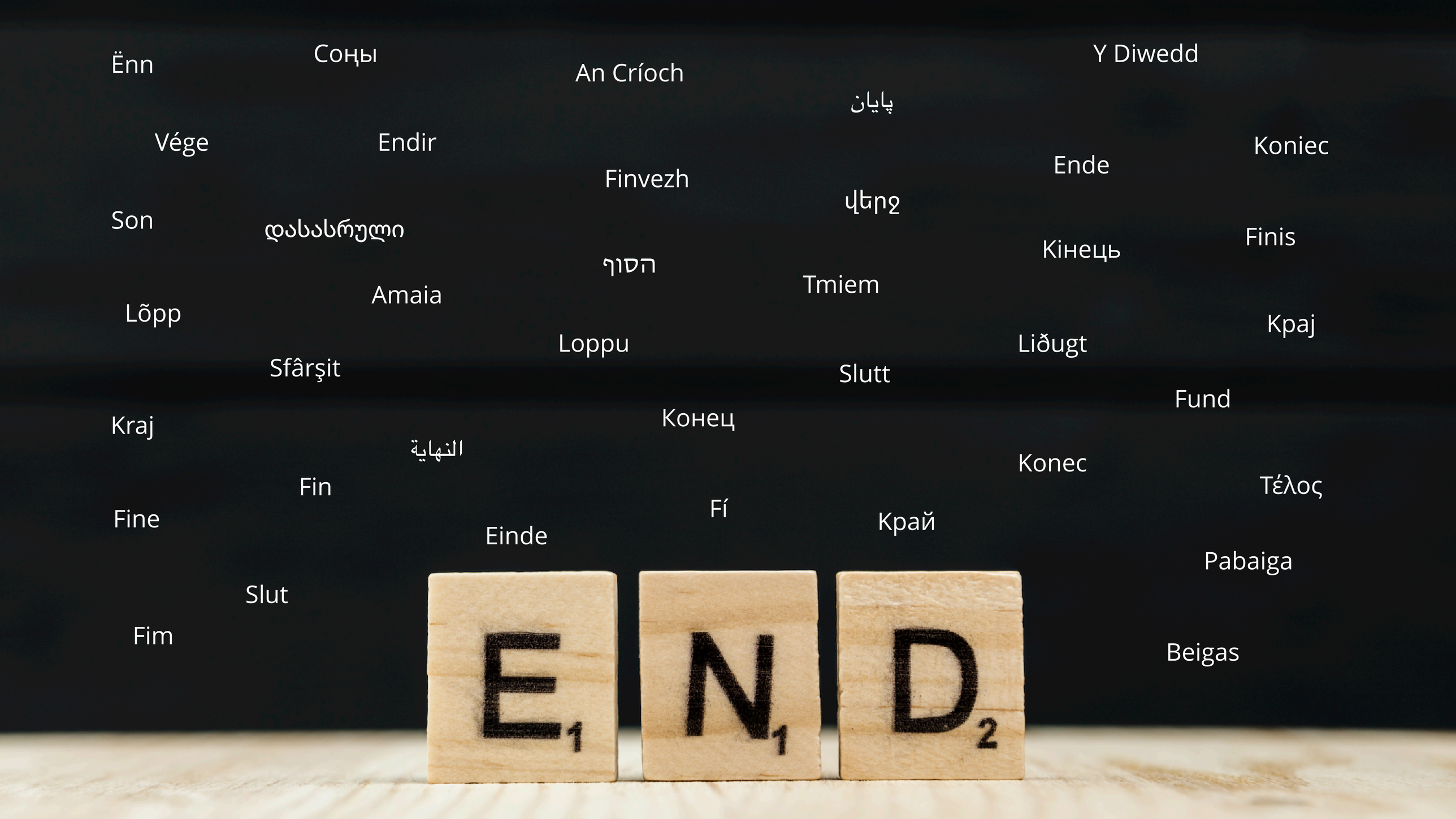
BGP Security E-learning Course

- Free online course
- Interactive, you can study at your own pace
- Practical lab environment and activities

academy.ripe.net/bgp-security/

RIPE NCC Certified Professionals

getcertified.ripe.net


Ënn
Соңы
An Críoch
پایان
Y Diwedd
Vége
Endir
Finvezh
վերջ
Koniec
Ende
Son
დასასრული
Finis
Кінець
הסוף
Amaia
Tmiem
Lõpp
Kpaj
Liðugt
Loppu
Sfârşit
Slutt
Fund
Kraj
Конец
النهاية
Konec
Τέλος
Fin
Fine
Fí
Край
Einde
Pabaiga
Slut
Fim
Beigas
E1 N1 D2

Copyright Statement

- The RIPE NCC Materials may be used for **private purposes, for public non-commercial purpose, for research, for educational or demonstration purposes**, or if the materials in question specifically state that use of the material is permissible, and provided the RIPE NCC Materials are not modified and are properly identified as RIPE NCC documents.
- Unless authorised by the RIPE NCC in writing, any use of the RIPE NCC Materials for advertising or marketing purposes is strictly forbidden and may be prosecuted. The RIPE NCC should be notified of any such activities or suspicions thereof.

Find the full copyright statement at:
https://www.ripe.net/about-us/legal/copyright-statement