

Revising the RPKI Validation Algorithm

draft-ietf-sidrops-rpki-validation-update

Job Snijders <job@fastly.com>

Terminology for reasoning about this

Assumed Trust

In the RPKI hierarchical structure, a Trust Anchor is an authority for which trust is assumed and not derived. Assuming trust means that violation of that trust is out-of-scope for the threat model.

Derived Trust

Derived Trust can be automatically and securely computed with **subjective logic. In the context of the RPKI, trust is derived according to the rules for validation of RPKI Certificates and Signed Objects.**

In other words

It is possible to define multiple deterministic validation algorithms for PKIs, like the RPKI.

Which algorithm is the right (or “correct”) algorithm is in the eye of the beholder.

The current algorithm is problematic

Defined in RFC 3779 section 2.3 and section 3.3; and RFC 6487 section 7.

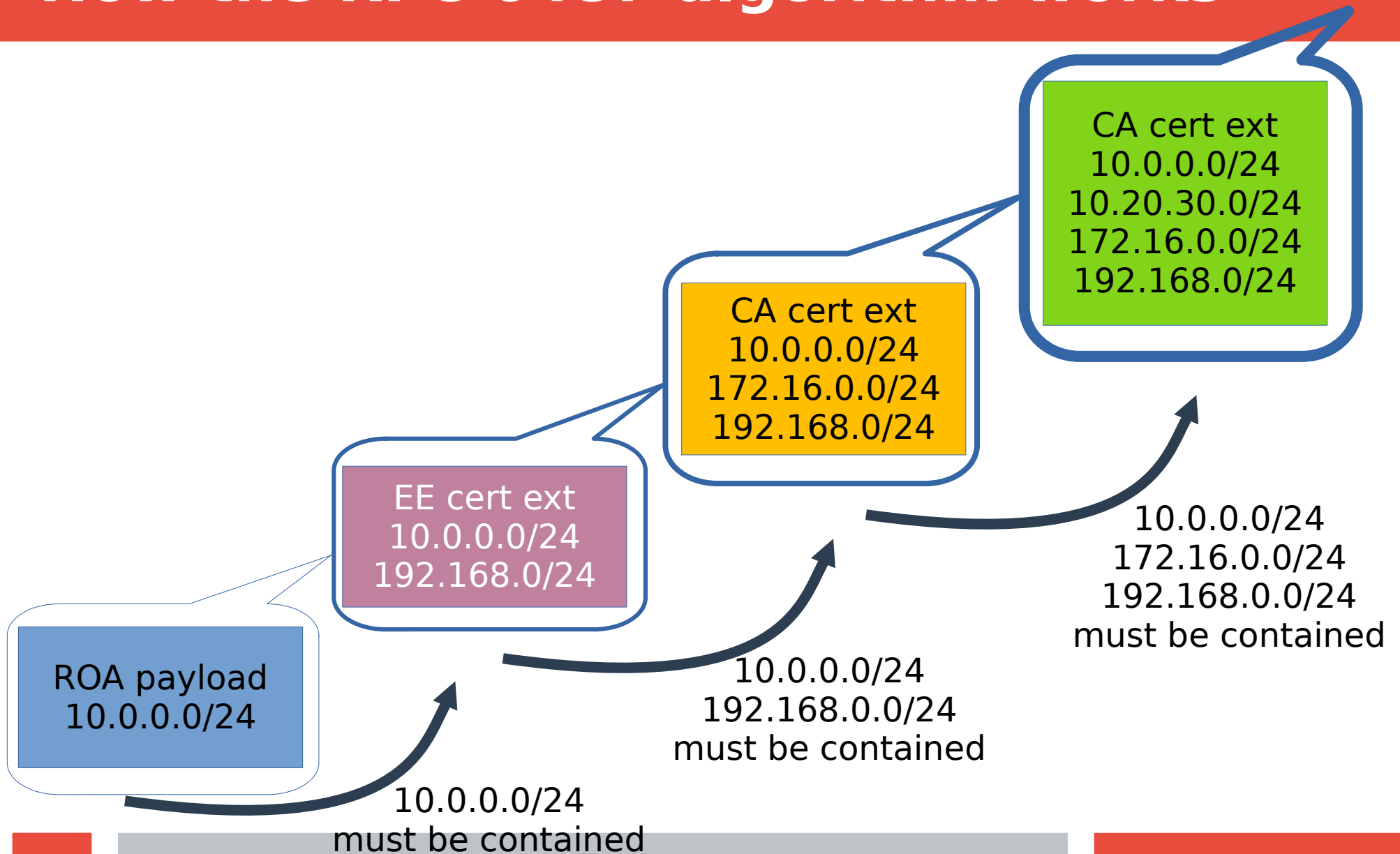
Number Resources *unrelated* to the ROA payload entry at hand also need to be contained (cumulatively).

Blast radius too big facing resource over-claiming.

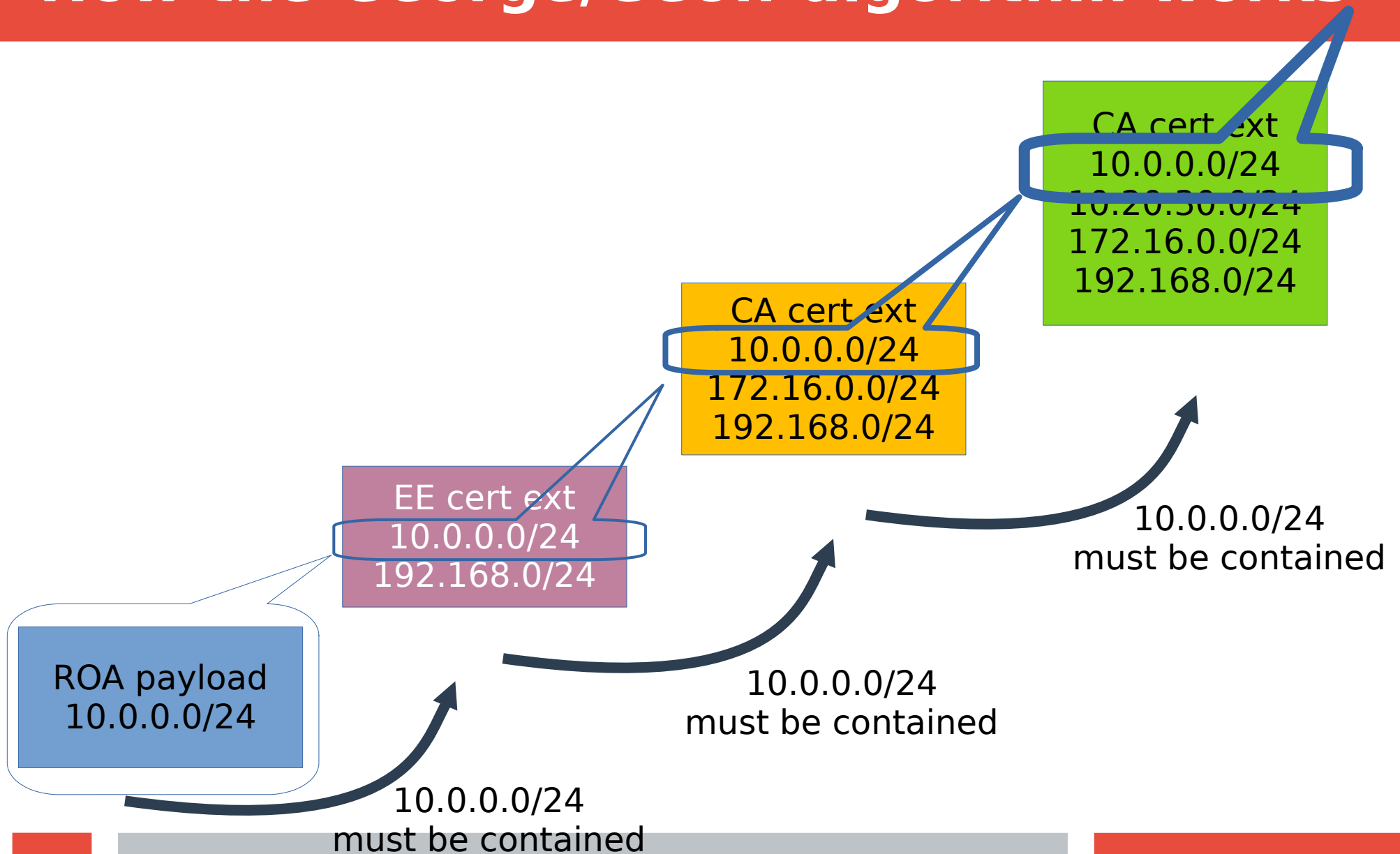
Lot of friction around inter-RIR/LIR transfers.

The 6487 outcome is disproportional in context of the RPKI

How the RFC 6487 algorithm works



How the George/Geoff algorithm works



Stepping through the algorithms

Starting point

ROA payload
10.0.0.0/24

EE cert ext
10.0.0.0/24
192.168.0/24

CA cert ext
10.0.0.0/24
172.16.0.0/24
192.168.0/24

CA cert ext
10.0.0.0/24
10.20.30.0/24
172.16.0.0/24
192.168.0/24

EE cert ext
172.16.0.0/24

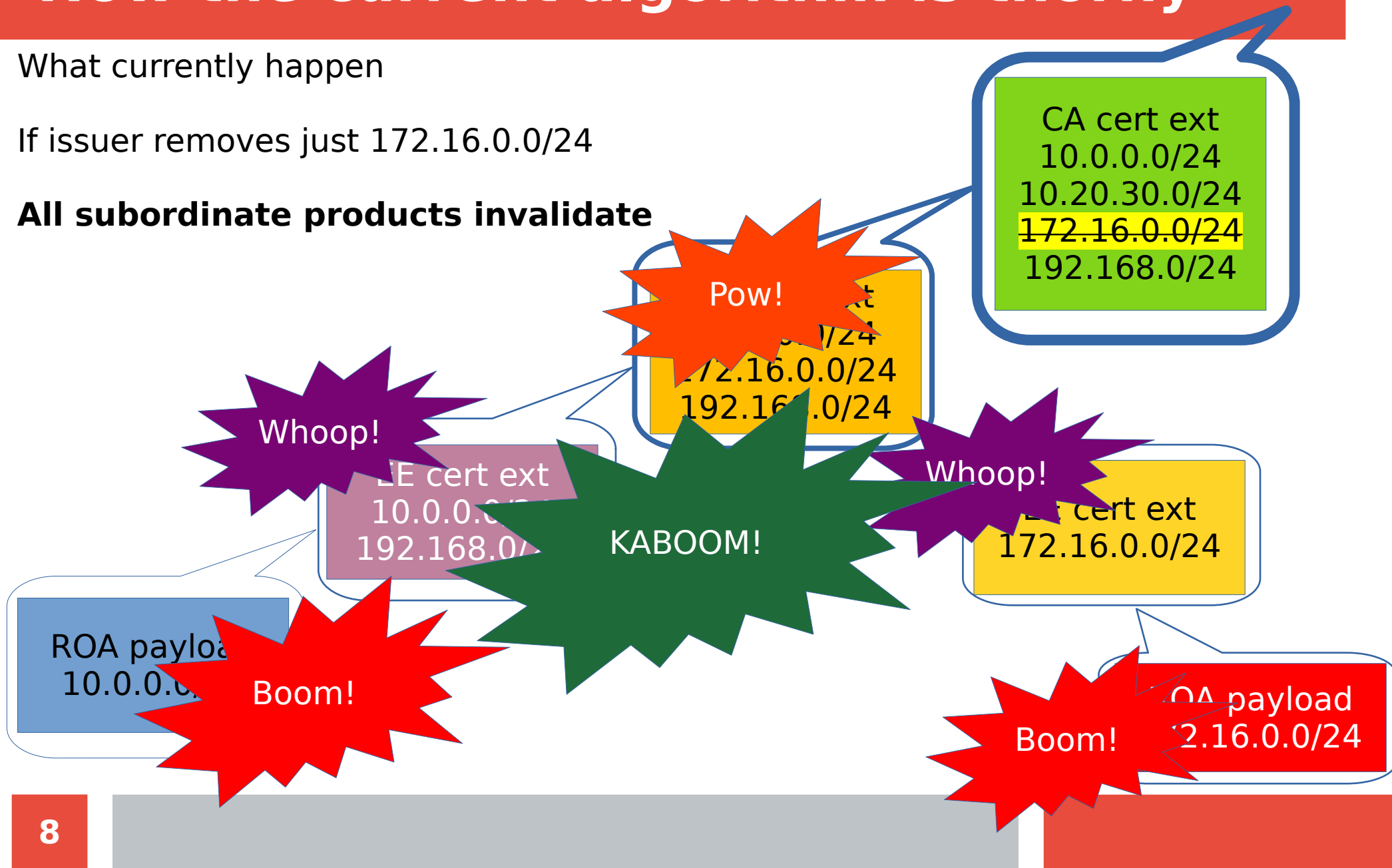
ROA payload
172.16.0.0/24

How the current algorithm is thorny

What currently happen

If issuer removes just 172.16.0.0/24

All subordinate products invalidate

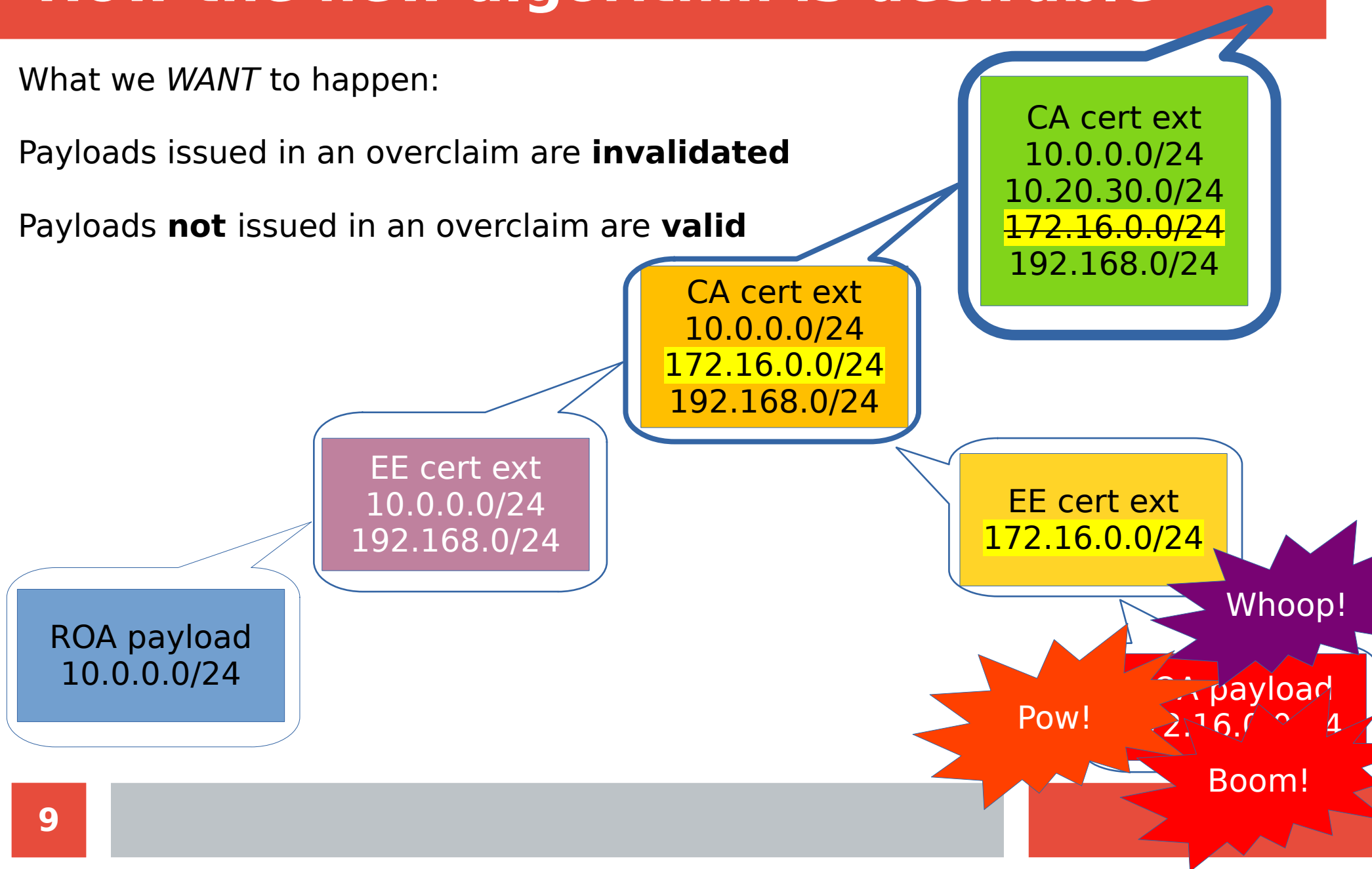


How the new algorithm is desirable

What we *WANT* to happen:

Payloads issued in an overclaim are **invalidated**

Payloads **not** issued in an overclaim are **valid**



The new algorithm is what we want

**This algorithm was proposed in RFC 8360
“Validation Reconsidered”**

**In summary: Number Resources *unrelated* to the
VRP entry at hand, do not need to be contained.**

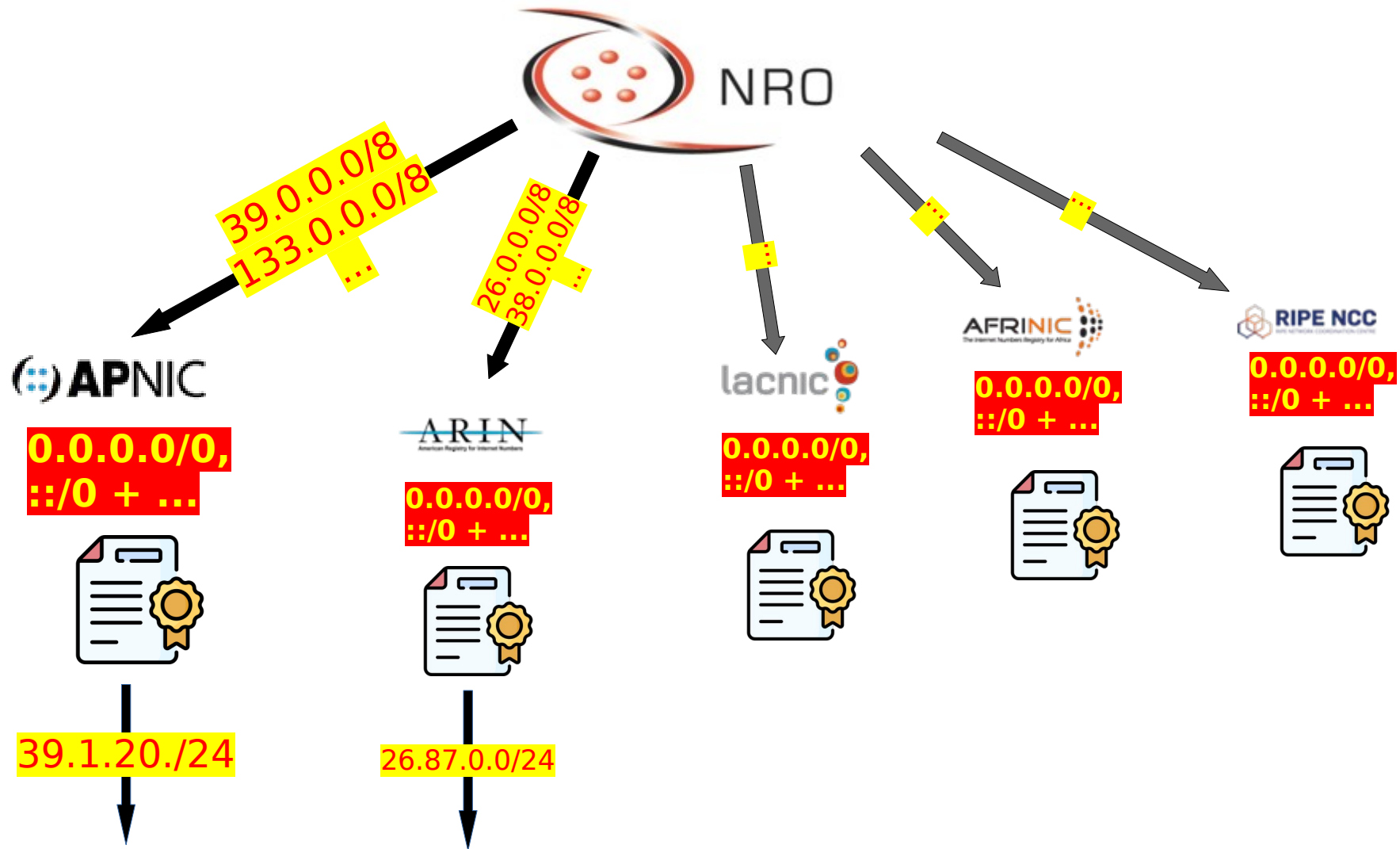
Blast radius is precise and limited.

***The new algorithm secures payloads how we
want.***

Why is the new algorithm important?

- 1) Less potential for issues with resource transfers (explained in slides 5 - 10)**
- 2) The new algorithm is needed for a single unified NRO-managed Trust Anchor!**

NRO Experimental Trust Anchor



Experimental NRO Trust Anchor

- 1) The NRO Trust Anchor imposes constraints on each RIR according to their current holdings**
- 2) The NRO Trust Anchor rechains to the existing RPKI Infrastructure**
- 3) This means at the top of the tree, RIR-issued certificates are “overclaiming”, *but pose no danger in and of themselves***

More information: <https://labs.apnic.net/nro-ta/>

RFC 8360 is undeployable

“Validation Reconsidered” was imagined to work via new policy identifiers where CAs and RPs do a complicated dance.

RFC8360 is under-specified, things are missing, but adding text won't solve the core issues.

The 8360 idea & algorithm are good, the execution plan is not feasible.

The path forward

Deprecate RFC 8360 & its code points

Update RFC 3779 & RFC 6487 to insert the new validation algorithm.

All implementation effort is with RP projects, CAs signers do not need to take action.

Validator projects seem to like this approach.

Next steps

- Please review [draft-ietf-sidrops-rpki-validation-update](#)
- Coding: implement the new validation algorithm in validators like *rpki-client*



Other work / recommended reading

draft-ietf-idr-bgp-sendholdtimer

draft-spaghetti-sidrops-rrdp-same-origin

draft-ietf-sidrops-avoid-rpki-state-in-bgp

draft-ietf-sidrops-rrdp-desynchronization

draft-ietf-sidrops-cms-signing-time

draft-harrison-sidrops-manifest-numbers

draft-snijders-constraining-rpki-trust-anchors

<https://datatracker.ietf.org/person/Job%20Snijders>