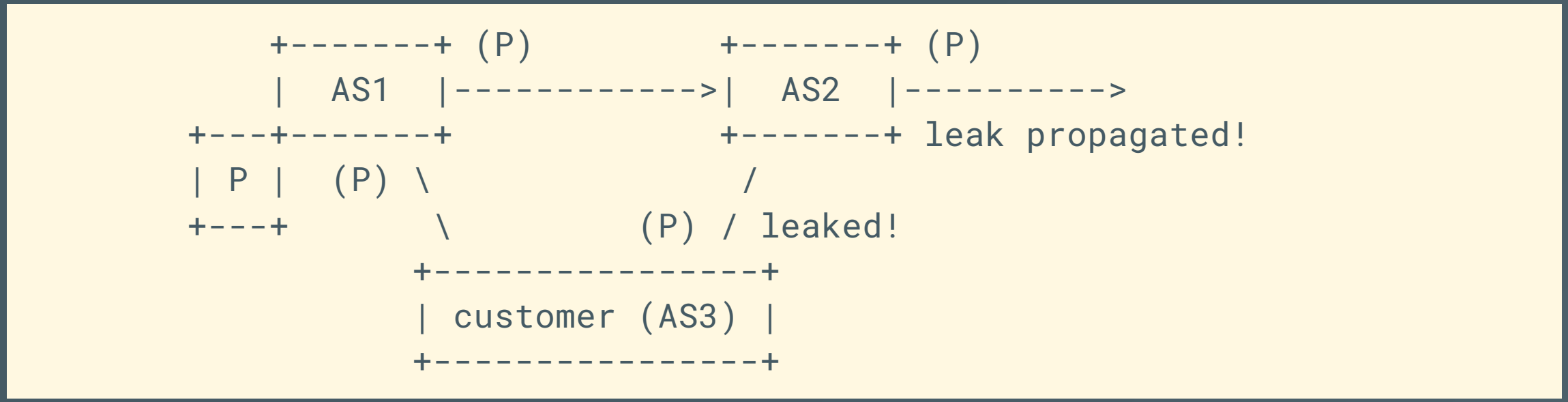
ASPA: RPKI-based AS_PATH verification

Ben Maddison
benm@workonline.africa

Background: BGP route leaks

> A route leak is the propagation of routing announcement(s) beyond their intended scope - RFC7908



Where's the harm?

Route leaks hurt *everyone*:

- Performance to the destination network is impacted by congestion or black holes
- The leaker's legitimate downstream networks are impacted by congestion upstream
- The leaker's connected networks (incl. IXPs) experience congestion because of the additional traffic being attracted
- The leaker incurs additional charges for transit utilisation

Where's the harm? (cont.)

- The origin's legitimate transit providers loose out on billable traffic
- Security and policy controls are bypassed
- NOCs everywhere try to diagnose problems that they don't have enough data to understand

Every AS that propagates the leak increases the blast radius

What does the solution look like?

Data describing the "intended propagation scope" of a BGP path that is:

1. Formulated in terms of data visible in BGP
2. Useful regardless of proximity to a leak
3. Strongly attributable and non-repudiate-able
4. Universally accessible

What does the solution look like? (cont.)

Good news!

If we can describe a *data structure* and *authorisation model* that fulfills #1 and #2, then the existing RPKI gives us #3 and #4 for free

:-)

Who gets to decide on "intended scope"?

- Prefix owner?
- Downstream AS?
- Upstream AS?
- Routing police?

Who gets to decide on "intended scope"? (cont.)

- Intuitively, a route has been leaked when no-one is paying the transit AS.
- Formalised in the "valley-free" model

Who gets to decide on "intended scope"? (cont..)

An observed `AS_PATH` is in agreement with intended routing policy when for each transit AS, either:

- the transit AS is authorised by the *sending* AS to announce the path upstream to non-customers; or
- the transit AS is authorised by the *receiving* AS to announce to it all the paths received from non-customers

ASPA RPKI signed object

- Authorisation by a *Customer AS (CAS)* of a *Set of Provider ASes (SPAS)*
- Based on RFC6488 object template
- CAS holder signs
- RP validates, aggregates, and sends to BGP speaker via RTR protocol

Object `eContent`

High level structure:

```
ASProviderAttestation ::= SEQUENCE {
     version [0]   INTEGER DEFAULT 0,
     customerASID  ASID,
     providers     ProviderASSet }

ProviderASSet ::= SEQUENCE (SIZE(1..MAX)) OF ASID

ASID ::= INTEGER (0..4294967295)
```

Object `eContent` - `version`

Familiar version construct. Nothing to see here.

```
version                 [0] INTEGER DEFAULT 0,
```

Object `eContent` - `customerASID`

AS number of the network providing and signing the authorisation.

Encoded as 32-bit integer.

```
customerASID        ASID,
```

Object `eContent` - `ProviderASSet`

- **Non-empty** set of authorised provider ASes
- No distinction between up/downstream authorisation
- `AS0` used to signal "transit-free". *Subject to change*
- **no longer AF-specific**

```
ProviderASSet ::= SEQUENCE (SIZE(1..MAX)) OF ASID

ASID ::= INTEGER (0..4294967295)
```

ASPA object processing

- ASPA objects are produced by RPKI CAs
  draft-ietf-sidrops-aspa-profile
- RPKI-RTR is (usually) how the data gets to the router
  draft-ietf-sidrops-8210bis
- ASPA verification algorithm operates on the data contained in the RTR payload (aka **VAP**).
  draft-ietf-sidrops-aspa-verification

BGP Route Processing

Each BGP path gets an `AS_PATH` verification state:

- **Valid**: all transit ASes appearing in the `AS_PATH` were verified by ASPA data
- **Invalid**: at least one transit AS in the `AS_PATH` is acting in contravention of its neighbors' ASPA authorisations
- **Unknown**: insufficient ASPA data exists to arrive at either Valid or Invalid

BGP Route Processing (cont.)

`draft-ietf-sidrops-aspa-verification` defines two algorithms:

1. **Algorithm for Upstream Paths**
   For paths received from non-transits (customers, peers, etc).
   The entire `AS_PATH` is expected to contain only *customer-to-provider* adjacencies

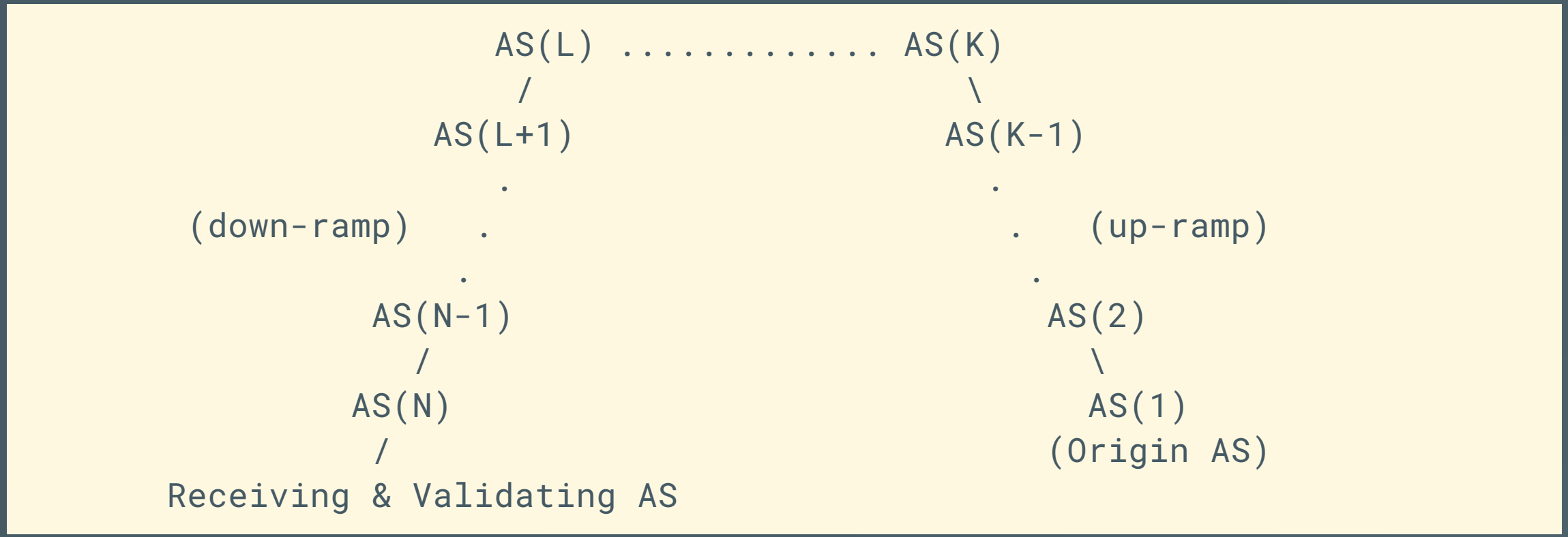
BGP Route Processing (cont..)

`draft-ietf-sidrops-aspa-verification` defines two algorithms:

2. **Algorithm for Downstream Paths**
   For paths received from transits.
   The `AS_PATH` is expected to contain:
   - An **up-ramp** of *customer-to-provider* adjacencies
   - A **down-ramp** of *provider-to-customer* adjacencies

BGP Route Processing (cont...)

Up-ramp / down-ramp visualisation



Alternatives?

- IRR data does not contain the necessary policy information (no `transit-via` in `aut-num`)
- Peerlock has similar semantics, however:
  - No crypto (in general)
  - Highly manual
  - Requires bug-free `AS_PATH` regex ;-)
- BGPsec solves a different problem - truthfulness of `AS_PATH`, not verification of routing policy

Benefits

Minimal information disclosure:

- no public assertions about who your peers or customers are
- compatible with non-disclosure obligations
- low change velocity for most operators

Benefits (cont.)

Incrementally deployable:

- Far-end verification: leaks are detectable several AS hops away
- A small number of published ASPA objects can make a large number of leaks detectable
- A small number of operators dropping ASPA "Invalid" paths can protect a significant part of the Internet

Benefits (cont.)

Well defined semantics:

- Orthogonal to other RPKI use cases: semantics of other objects don't change
- Compliments ROV, BGPsec, etc.
- Sensible policy granularity: policy is described at the AS level, no sessions or prefixes[*]

[*]: *See OTC Attribute RFC9234 for prefix-granularity detection*

Current Status - IETF

- draft-ietf-sidrops-aspa-profile and draft-ietf-sidrops-aspa-verification currently in WGLC.
  - Mostly complete and stable
  - Discussion ongoing about how "transit-free" should be represented
- draft-ietf-sidrops-8210bis was awaiting RFC publication - needs a revision to remove per-AFI data structure

Please review!

Current Status - Implementations

- CA implementations - Krill, RIPE NCC (pilot)
- RP implementations - `rpki-client`, Routinator, RPSTIR2, StayRTR
- Tooling and testing - `rpkimancer`, various others
- BGP speaker implementations - `openbgpd`, NIST BGP-SRx

Still missing commercial NOS vendors

Operator involvement

Operators should be planning for ASPA now:

- Consider whether the verification algorithm is compatible with your current routing policy?
- Start talking to your peers, customers and transits about deployment
- Ask your router vendors about their roadmap

FIN