Revisiting RPKI LTAM (inertia may be your friend)

Stephen Kent

BBN Technologies

Local TA Management: Why

- There are times when an operator wants to assert ownership of a prefix (or an AS #) in a local context
- In such cases it would be nice to be able to make these assertions, locally, without having RPKI/ BGPSEC software complain (to you, as the operator)
- The obvious case is use of RFC 1918 address space
- If an assertion about an IANA reserved address "escapes" the local context, it will be rejected by operators who make use of the RPKI, so other nets ought not be adversely affected

Another Local TA Motivation

- A nation might worry that some entity in the resource allocation hierarchy could (accidentally or maliciously) revoke a certificate for critical infrastructure resources (in that nation, or elsewhere)
- A nation can protect nets within its administrative jurisdiction against such mishaps IF it can direct internal nets to rely on a national authority for RPKI data for these critical infrastructure resources
- If the country could externally declare the ROA data for its ISPs, that would be even better (subject to appropriate controls). LTAMv1 could not do this.

Why LTAM v2?

- LTAM has been presented at SIDR meetings since IETF 75 (July, 2009) but little feedback has been received
- It would work well for the RFC 1918 use case
- It probably would work well at the IANA & RIR level for the 2nd use case (local override of repository system)
- It would not work so well for ISPs that delegate address space, hole punching conflicts with ISP ROAs
- It would not support the external declaration capability for countries trying to protect their ISPs ROAs
- So, we were motivated to revisit the LTAM design

A New Model

- We decided to focus only on ROAs, since changes to the RPKI that adversely affect ROAs are of interest to the associated INR holders (and potentially, to RPs)
- We anticipate this model can be extended to cover router certificates too, for BGPSEC
- The model has three elements
  - INR holders detect when their own ROAs no longer validate or are in “competition”
  - INR holders publish “external” info to protect their ROAs
  - RPs detect “adverse” ROA changes, check external info to decide if the change is OK, or revert to old data

Adverse ROA Changes

- ROA Whacking – A ROA is whacked when it becomes invalid due to any action by a CA (or publication point maintainer) along the path between the ROA EE certificate and a trust anchor. Whacking includes ROA certificate revocation, CA certificate revocation, CA certificate 3779 extension changes, removal of a ROA from the RPKI repository, etc.
- ROA Competition – A new ROA "competes" with an exiting ROA when the new ROA is issued by a different entity, points to a different ASN, and contains the same or a more specific prefix. Competing ROAs are legitimate in some cases, but illegitimate overlaps represent a way to divert traffic.

Design Criteria

- Each INR holder decides whether to “protect” its ROAs
- Each RP decides whether to invoke new mechanisms to detect ROA whacking/competition
- Efficient ROA whacking and competition detection by RPs and INR holders
- RP access to previously validated ROA data
- Accommodate transfers, MOA, key rollover, etc.
- Work in the face of outsourced CA & pub point mgmt.
- Any new RPKI objects conform to RFC 6488

Self-Monitoring of ROAs

- To first order, every INR holder is also an RP
- Every RP downloads all changed RPKI data at least daily
- An INR holder could easily configure its own ROA data and use that data to check their status
- The RP software should provide info to the INR holder to identify why/where ROA validation fails
- The INR holder should contact the indicated party (Ghostbusters!) to resolve the problem

Publishing “External” Data

- If an error has caused a ROA to become invalid (or missing), RPs may want to ignore the problem, for a little while, to give the INR holder a chance to fix the problem (before treating the ROA as not valid)
- But, how does an RP know whether an adverse change to ROA data is sanctioned by the INR holder?
- The INR holder needs a way to signal to RPs when the INR holder makes changes to its ROAs
- This data needs to be external to the RPKI repository system, to be an independent assertion about ROAs

The Tough Case

- Today, most INR holders who have certificates and ROAs make use of outsourced CA and publication point management offered by the INR holder's RIR
- If an INR holder fears that the RIR may have been compelled to whack his ROA (e.g., by law enforcement) then he can't rely on his ability to change data within his publication point to fix a problem when it is detected
- An INR holder could publish status data where it is not under the control of the CA/publication point maintainer
- RPs need to be able to verify the integrity of the file

What Could RPs do?

- If an RP wants to give INR holders an opportunity to fix problems when adverse ROA changes are detected, the RP can use the external INR data to add some inertia to the change process
- An RP that elects to adopt this approach ought not be burdened by this capability
  - It ought to be easy to detect adverse ROA changes
  - Additional data retained by RPs should be small
  - External data ought to be fetched ONLY if it has changed
- It seems possible to meet these criteria by adding one new RPKI object, and a simple external file format

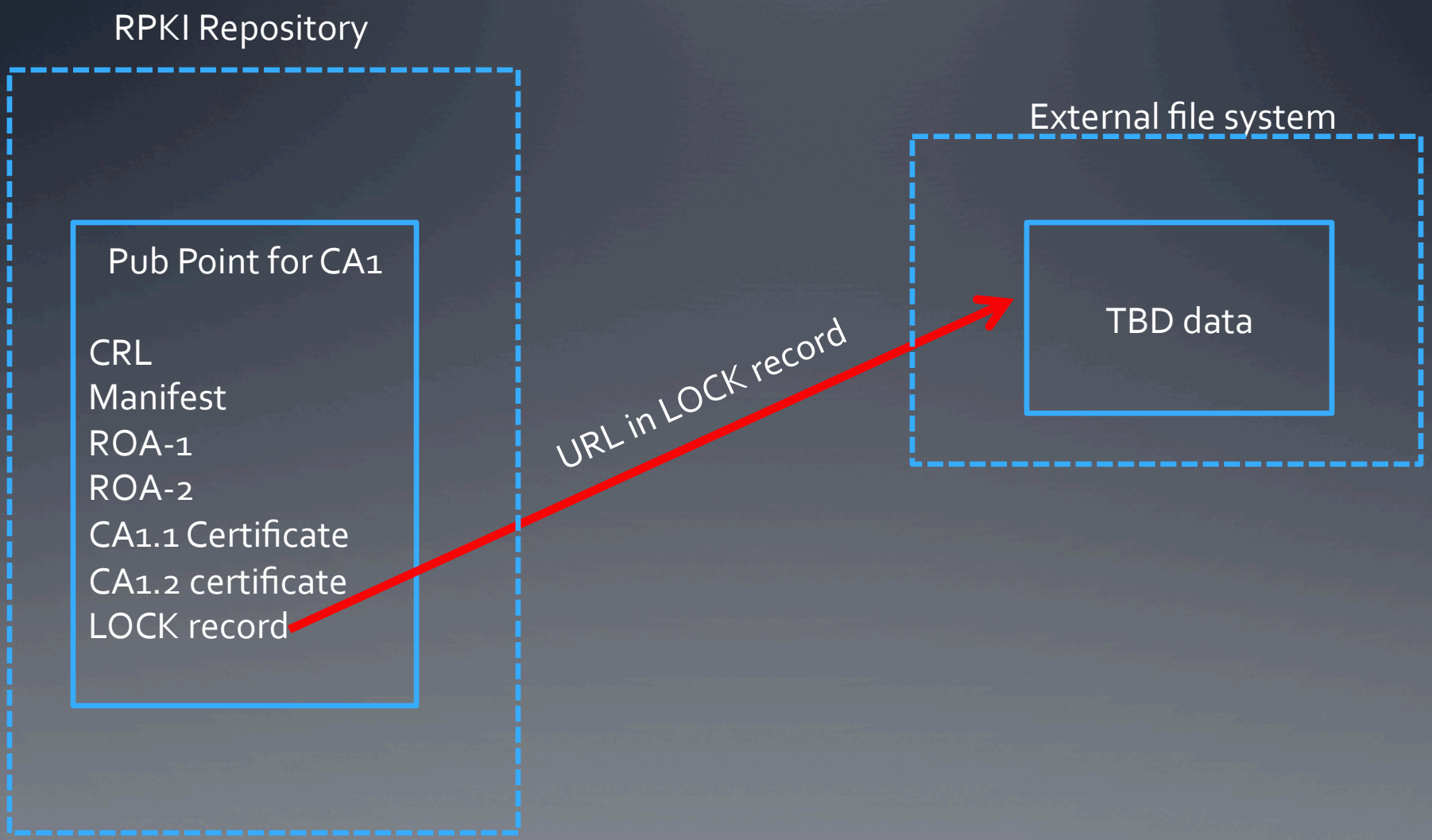
Solution Outline - Data

- An INR holder may optionally add a new RPKI signed object (LOCK) to its publication point, to refer to external data
- The LOCK object contains a URL pointing to the external data file, and an integrity check for that data
- The content of the external data file is still under discussion (but we have a candidate name: Internet Numeric Resource Reference Data = INRRD)
- Each RP that elects to make use of this data will need to maintain a local database of ROA data asserted by each INR holder, using this mechanism

Solution Outline - Processing

- When an INR holder makes a change to its ROA data that would appear to RPs to be adverse, it must update its INRRD file first to reflect this change
- RPs that choose to support anti-whacking scan changed ROA data for adverse changes
- An adverse change is accepted by an RP if the INRRD file corroborates the change; otherwise the RP may elect to revert to previously validated ROA data for this INR holder

Graphic Details



More Stuff to Do

- Document why this mechanism is adequate, from a security perspective
- Document procedural details for INR holders and RPs with respect to transfers, delegation, algorithm change, key rollover, etc.
- Document algorithm for detecting ROA whacking and competition
- Document algorithm for reporting who whacked a ROA or has a competing ROA (for INR holder use)
- Select format for INRRD file, e.g., ASN.1 vs. JSON)

QUESTIONS ?