RPKI Out-Of-Band Setup Protocol

Rob Austein <sra@hactrn.net>
Randy Bush <randy@psg.com>
Michael Elkins <Michael.Elkins@sparta.com>
. . . and a lot of help from our friends

IETF 88
Vancouver
November 2013

Purpose

Encapsulate BPKI public keys, subject names, and service URLs needed to set up RPKI provisioning (RFC 6492) and publication (draft-ietf-sidr-publication) protocols in a simple interoperable format.

What This Protocol Deliberately Leaves Out

- How these messages are exchanged is deliberately unspecified. USB stick, PGP-signed email, HTTPS, T-shirt printed with QR code, carrier pigeon, ...
- Receiver must authenticate and check integrity of messages, but how receiver does this is also deliberately unspecified.

History

- Earliest setup experiments just passed around BPKI certificates and URLs. Mistakes were frequent and almost inevitable.
- Tokyo RPKI workshop (January 2010) hit upon idea of a simple encapsulation so that each step in the protocol would involve sending exactly one well-formed message with labeled fields.
- Other RPKI CA engine implementors implemented provisioning portion of the protocol to simplify inter-operation.
- At this point, our setup protocol has become the de facto standard for provisioning protocol setup.
- Review of user experience concluded that protocol semantics were OK but syntax was unnecessarily confusing.

Where We Are Now

- draft-austein-sidr-rpki-oob-setup describes a cleaned-up version of the protocol.
- Semantics are unchanged, only syntax is different from what we're using now.
- Converting existing implementations to the new syntax should be easy.
- Pre-00 versions of the draft added features to the protocol, but discussion among implementors concluded that none of the new stuff was worth changing protocol semantics, so we dropped it.

The Setup Minuet

1. Child(Alice)→Parent(Bob): “Hi, I’m Alice, here’s my BPKI key, and I’d like to get RPKI resources from you.”
2. Parent(Bob)→Child(Alice): “OK, I’m Bob, here’s my BPKI key, I’m going to call you Alice-17, you can contact me using the provisioning protocol at URL http://bob.example/alice-17, and maybe Carol can help if you’re looking for a repository to use.”
3. Publisher(Alice)→Repository(Carol): “Hi, I’m Alice, here’s my BPKI key, I’d like to publish in your repository, Bob sent me.”
4. Repository(Carol)→Publisher(Alice): “OK, here’s my BPKI key, you can publish your stuff under URI rsync://carol.example/rpki/bob/alice, and you can contact me using the publication protocol at URL http://carol.example/bob/alice.”

Who Must Do What

- Bob doesn't have to accept Alice as a child.
- Carol doesn't have to accept Alice as a publisher.
- Alice doesn't have to use Carol as a repository.
- Bob can call Alice anything Bob wants, the name Alice gives to Bob is just a hint. This matches expected RFC 6492 behavior.
- If Bob and Carol are the same entity, we call it a "publication offer," otherwise we call it a "publication referral;" referrals include an authorization token to support hierarchical repository structures.

BPKI Keys & Certificates

- “BPKI keys” in the above description are really self-signed X.509 BPKI certificates, for historical reasons given how the protocol evolved. We could have used PKCS#10, but we didn’t, and we see no obvious benefit to changing this now.
- Details of exactly how receivers use incoming BPKI keys are implementation specific, but probably involve some form of cross-certification.
- Recommended approach: Receiver checks self-signature, then extracts public key and subject name and cross-certifies under receiver’s own BPKI root, using a Basic Constraints extension with cA = TRUE and pathLenConstraint = 0.

Ready For Standardization?

- This is already a de facto standard, developed outside the IETF.
- We think it's ready to be documented as an interoperable standard.
- Does the WG agree?
- Does the WG want this as a work item?

Thanks To. . .

- Warren Kumari, First Guinea Pig.
- The participants in the 2010 Tokyo workshop, who told us we needed this protocol.
- The other RPKI CA implementors, for making this work with their engines.
- All of our beta testers, for helping us get the semantics right.
- Leif Johansson, for telling us to fix the syntax.
- Everyone who reviewed the pre-00 draft.
- Our sponsors, who paid for all this entertainment.

Questions?