Algorithm Transition for the RPKI

Steve Kent
Roque Gagliano
Sean Turner

Algorithm Transition Document

- Target is a standards track RFC from SIDR
- A plan for algorithm agility for the RPKI, requested by Tim Polk
- Same algorithm key rollover already covered by the Key Rollover I-D
- The document describes
  - what CAs have to do to effect algorithm transition
  - what RPs can expect during algorithm transition

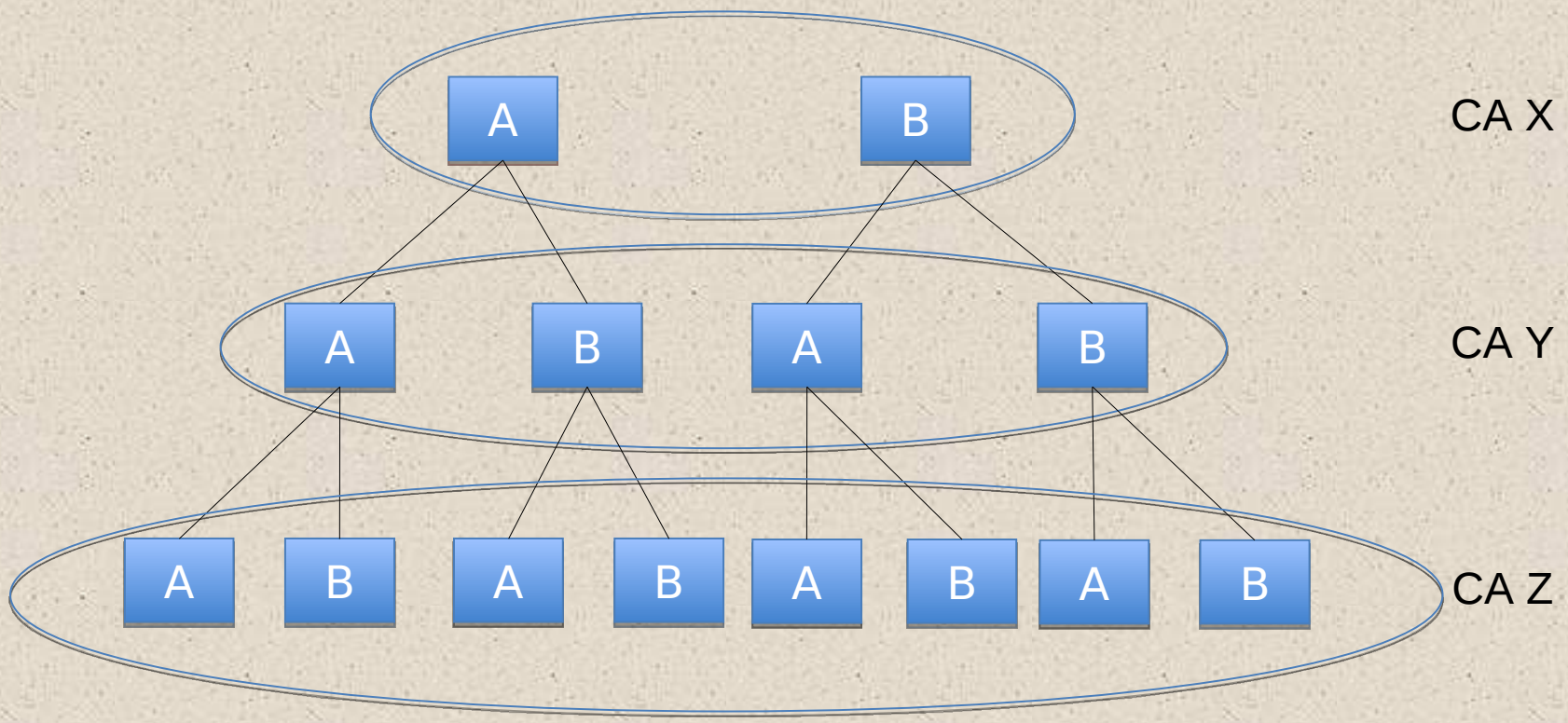
What Happened Since Maastricht

- draft-rgaglian-sidr-algorithm-agility-00 issued
- This I-D reflects the Maastricht briefing, plus some refinements
- Steve just realized that full, mixed-algorithm certificates are not likely to be practical, due to repository growth concerns, and they do not seem to be necessary
- And, if no mixed-algorithm certificates are issued, then we can't support a Laissez-faire algorithm transition model

Terminology

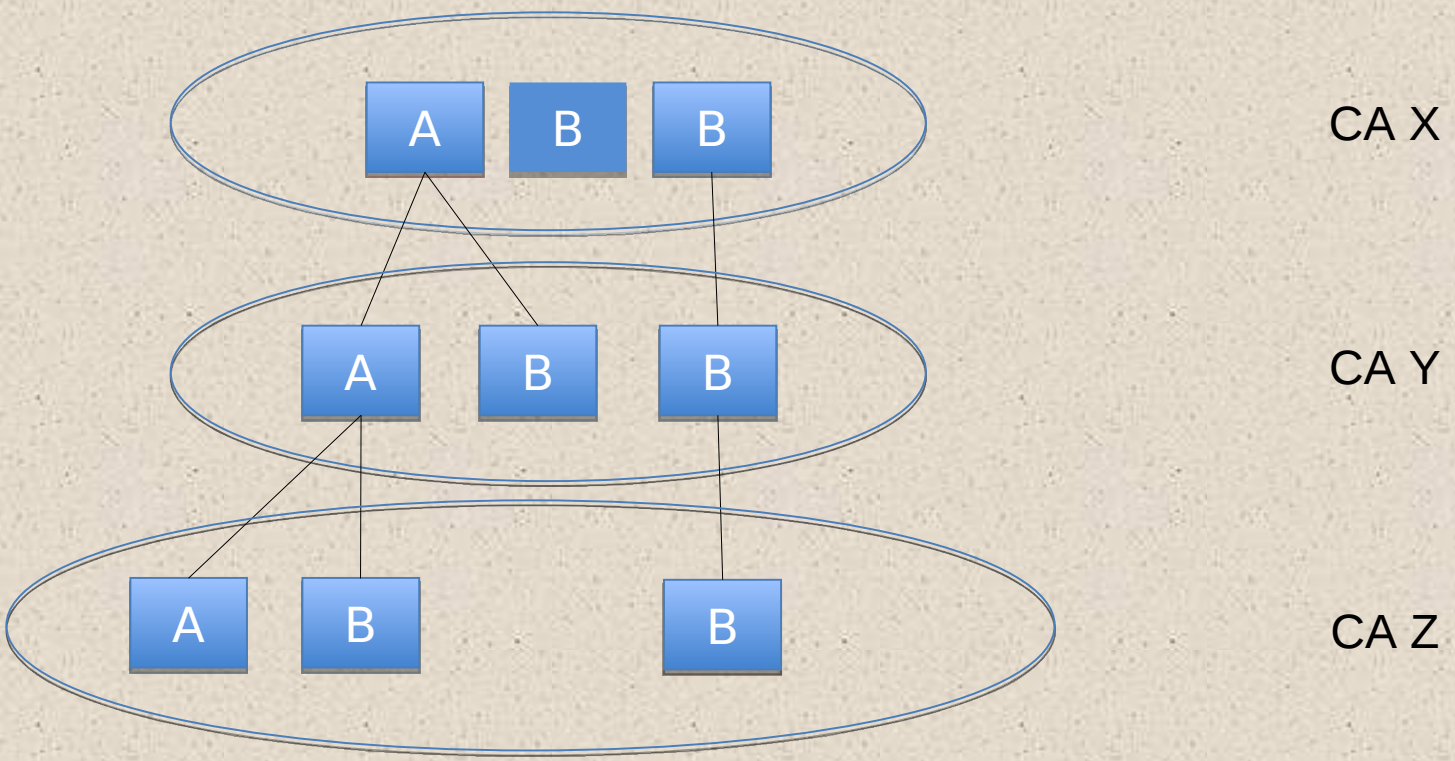
- Algorithm A - the current signature and hash algorithm suite
- Algorithm B - the next signature and hash algorithm suite

Full Mixed Algorithm Certificates



We get exponential repository growth IF we require every CA to issue certificates to subordinate CAs with an algorithm A key and an Algorithm B signature! (The depth of the repository is probably no more than 5-6 in some places, but $2^6 = 64$!)

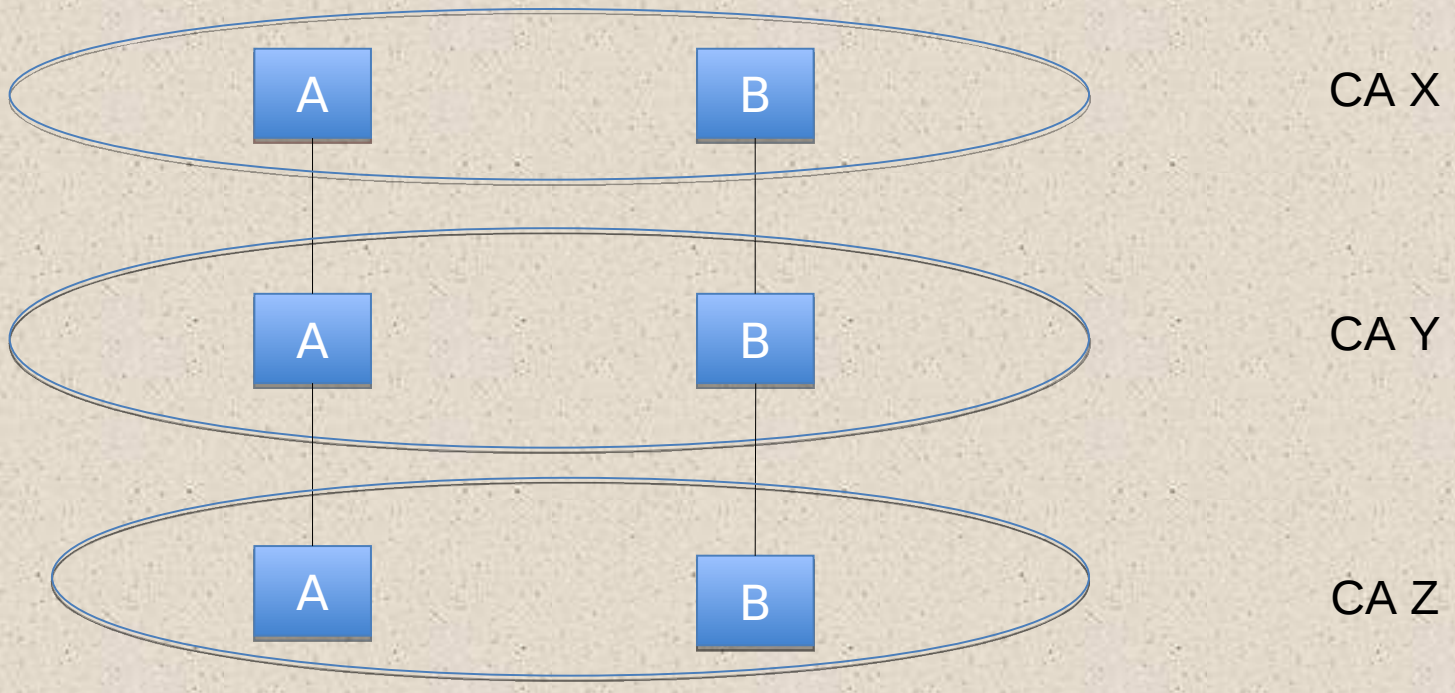
Partial Mixed Algorithm Certificates



Here a CA issues certificates with Algorithm B keys under an Algorithm A or Algorithm B signature, but will not issue a certificate with an Algorithm A key under an Algorithm B signature. Also, a CA never signs a certificate using Algorithm B under a CA whose Algorithm B certificate has a direct Algorithm A ancestor. This limits directory growth to 3X.

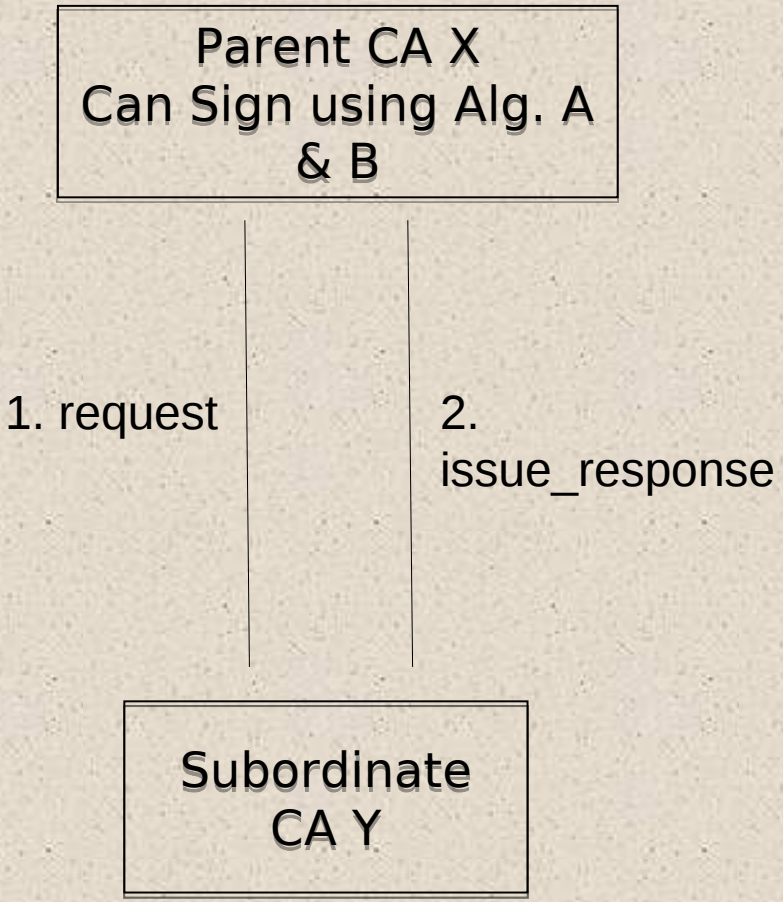
Uniform Algorithm Certificates

CA X's parent has to issue an Algorithm B certificate to CA X before that CA can issue an Algorithm B certificate to CA Y



If each CA issues certificates that use the same algorithm for signing and for the key in the certificate, then directory growth is at most 2X.

Signing Algorithm Selection problem



How should CA X respond to a request when it supports more than one algorithm suite?

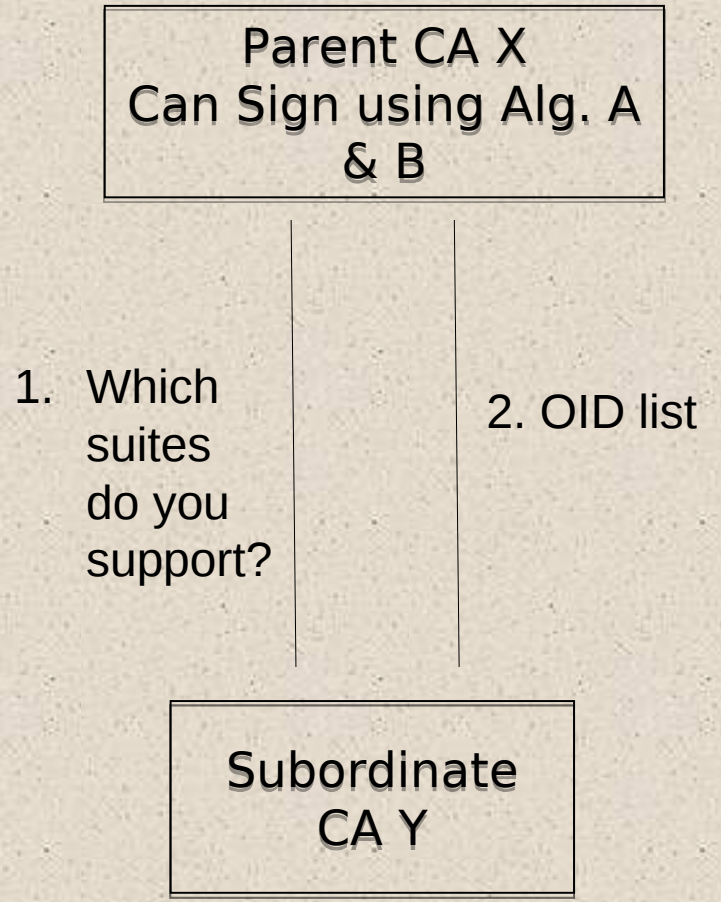
1)We propose the addition of an attribute to the “request” message: "signing_algorithm=OID”

Also, need to add an error response: 1401 request - request - signing algorithm not supported.

the requester has complete control and we keep the relationship: one request, one certificate issued

If CA X has separate “contact points” for Algorithm A vs. Algorithm B, then this may not be needed

Capabilities Exchange



Algorithm migration might be easier if one could query a CA about what algorithms it supports

With this mechanism, the subordinate CA can be configured to effect migration phases without (or with less) manual intervention

Alternatively this could be implemented by trial and error, if error 1401 is supported

Repository Management

- The transition process requires CAs to publish signed products under Algorithms A & B
- During “CA Algorithm B Ready” there may be only a few products (CA certificates) under Algorithm B, so the duplication is not complete
- By “CA Algorithm B Go” a full, duplicate product set exists, 2X repository load (3x if we allow partial, mixed algorithm certificates)

Relying Parties

- This design provides RPs with signed products under the current and new algorithms for a while, because we assume that it will take a while for ALL RPs to be able to make the transition
- This imposes a burden on CAs to maintain parallel signed product sets from “CA Algorithm B Go” until “Algorithm A Twilight”
- We need to decide what an RP should do if it fetches both product sets, and encounters mismatches

Observations

- This is a brand new document, and thus needs review and comments from the WG
- There are some significant differences between algorithm transition and key rollover
- There will be repository growth, but it is manageable if we limit or prohibit mixed algorithm certificates
- Still need to work in details of key rollover taking place during algorithm transition
  - key rollover results in two sets of signed products, but only briefly, and locally, so this is probably OK

Questions

- Should the WG request the modification of the provisioning protocol to include a signing algorithm attribute and a new error code for “algorithm not supported”
- Should the WG request the modification of the provisioning protocol to include a mechanism to query the list of signing algorithms supported by a parent CA?
- Do we really need to support a Laissez-faire algorithm transition model?

Algorithm Transition Details

- Slides beyond this one will not be presented, but are included for background

CA & RP Transition Phases

Algorithm Transition Milestones (1/2)

- Steady state (using algorithm A)
- CA Algorithm B Ready – all CAs are ready to process a certificate request for a certificate containing an Algorithm B key, signed under Algorithm B (or Algorithm A)
- CA Algorithm B Set - all CAs are ready to issue certificates under Algorithm B
- CA Algorithm B Go – all CAs reissue all signed products under Algorithm B (and continue issuing under Algorithm A)

Algorithm Transition Milestones (2/2)

- RP Ready Algorithm B – all RPs are ready to process signed products using Algorithm B
- Twilight Algorithm A – CAs may stop issuing signed products using Algorithm A, and RPs may cease validating signed products under Algorithm A
- EOL Algorithm A – CAs no longer issue signed products using Algorithm A, and RP reject any signed product under Algorithm A

Relation to Key Rollover

- Like key rollover, algorithm transition relies on the AIA extension in a certificate pointing to the directory containing the parent certificate and CRL files
- Manifest is pointed to by SIA in CA certificate
- Like key rollover, a CA effecting algorithm transition will reissue subordinate CA certificates unilaterally
  - reissue under old algorithm suite, initially
  - reissue under new algorithm suite, later
  - both old and new will co-exist for a while

Repository Structure Dependencies

- Algorithm transition design must accommodate file name conventions in draft-ietf-sidr-repos-struct-xx.txt
  - file name for a CA certificate is derived from the public key in that certificate
  - file name for a CRL is derived from public key of the CA that issued the CRL
  - file name for a manifest is derived from the public key of the CA that issued the manifest
  - directory names are “arbitrary” acquired from SIA/AIA pointers
- These conventions determine which files are overwritten and which persist, during algorithm transition

Steady State

- All CAs and RPs are using the “current” algorithm suite
- IESG approves the “next” algorithm suite and publishes this as a revision of the RPKI algorithms RFC
- This revised document also establishes milestone dates for the transition process
- This signals the beginning of the transition

CA Algorithm B Ready

- As of this date, all (non-leaf) CAs MUST be accept a request from a child CA to issue a certificate containing an Algorithm B key, which will, be signed using Algorithm B
- This milestone allows CAs to begin generating algorithm B keys and getting certificates with algorithm B signatures
- No RPs are required to process these certificates, so this is a testing capability

CA Algorithm B Go

- By this date, every CA MUST re-issue all of its signed product set signed under Algorithm B
- This is the first time that a CA is required to issue & publish certificates (CRLs, manifests, etc.) under Algorithm B
- Still no requirement for ANY RP to accept these signed products, but RPs can test their ability to process Algorithm B products from all CAs

RP Algorithm B ready

- By this date, all RPs MUST be prepared to process signed material issued under Algorithm B
- Both Algorithm A and B products have been available, in parallel, since the previous milestone

Algorithm A Twilight

- By this date, a CA MAY cease issuing signed products under Algorithm A
- Also, after this date, an RP MAY cease to validate signed materials issued under Algorithm A
- This milestone marks the end of Algorithm A, as no CA can rely on RPs accepting products signed under that Algorithm, and no RP can rely on a CA to issue products under A

Algorithm A EOL

- As of this date every CA MUST NOT generate certificates, CRLs, or other RPKI signed objects under Algorithm A
- Also, after this date, no RP should validate any certificate, CRL or signed object using Algorithm A
- This marks the end of Algorithm A, and the transition to a Steady State